

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 September 2001 (27.09.2001)

PCT

(10) International Publication Number
WO 01/71876 A1

(51) International Patent Classification⁷: **H02H 3/05**,
H03K 19/003

(74) Agents: **LEBOVICI, Victor, B. et al.**; Weingarten,
Schurgin, Gagnebin & Hayes LLP, Ten Post Office Square,
Boston, MA 02109 (US).

(21) International Application Number: PCT/US01/08884

(22) International Filing Date: 20 March 2001 (20.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/191,602 23 March 2000 (23.03.2000) US
09/716,583 20 November 2000 (20.11.2000) US

(71) Applicant: **INTELLITECH CORPORATION**
[US/US]; 70 Main Street, Durham, NH 03824 (US).

(72) Inventors: **RICCHETTI, Michael**; 54 Cathedral Circle,
Nashua, NH 03063 (US). **CLARK, Christopher, J.**; 22B
Cedar Point Road, Durham, NH 03824 (US). **DERVI-**
SOGLU, Bulent, I.; 495 Sleeper Avenue, Mountain View,
CA 94040 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,
CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM,
HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK,
LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX,
MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL,
TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

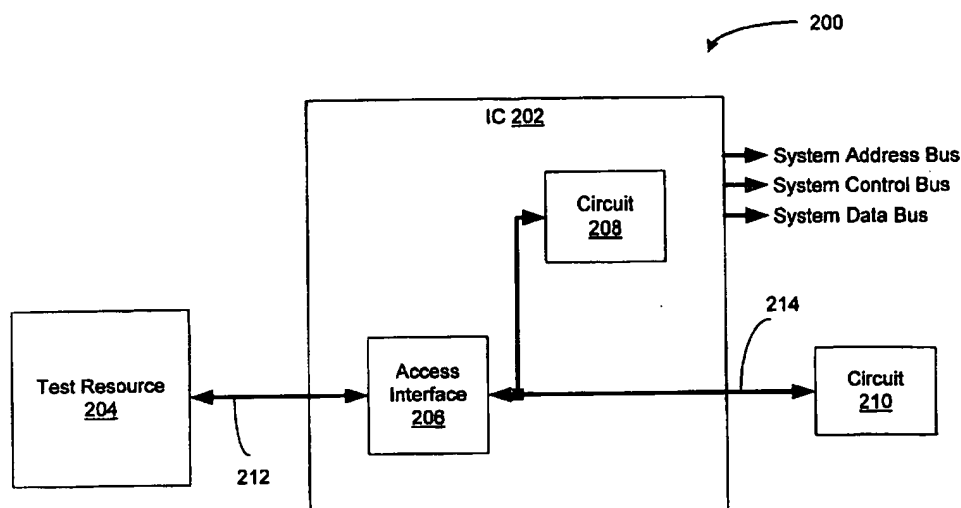
(84) Designated States (*regional*): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,
CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a
patent (Rule 4.17(ii)) for all designations*

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR PROVIDING OPTIMIZED ACCESS TO CIRCUITS FOR DEBUG, PROGRAMMING, AND TEST



(57) Abstract: An access interface (206) for accessing electrical nodes of an electronic circuit (208) for programming, testing, and debugging the electronic circuit (208). The access interface (206) includes a protocol generator and a data generator that may be programmed to apply control and/or data sequences directly to the electronic circuit. The access interface (206) performs operational commands based upon a plurality of states included in a programmable state machine. By suitably programming the protocol generator, the data generator, and the state machine, electrical nodes of the electronic circuit can be accessed in reduced time using a reduced number of operations. The access interface (206) is controlled by a test resource apparatus (204), which communicates with the electronic circuit (208) connected to the access interface (206).

BEST AVAILABLE COPY



— as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations

— with amended claims

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

-1-

TITLE OF THE INVENTION

METHOD AND APPARATUS FOR PROVIDING OPTIMIZED ACCESS TO
CIRCUITS FOR DEBUG, PROGRAMMING, AND TEST

5

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims the priority of U.S.
Provisional Patent Application No. 60/191,602 filed March
23, 2000, entitled METHOD AND APPARATUS FOR PROVIDING
10 OPTIMIZED ACCESS TO CIRCUITS FOR DEBUG, PROGRAMMING AND
TEST.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR
DEVELOPMENT

15

N/A

BACKGROUND OF THE INVENTION

The present invention relates to apparatus and
methods for accessing electrical nodes of electronic
20 circuits for programming, testing, or debugging purposes.

One way of providing access to electrical nodes of
Integrated Circuits (ICs) is commonly referred to as "scan
testing," which typically involves serially shifting
digital data into state elements included in an IC to
25 apply logic levels to selected nodes as test stimuli, and
serially shifting digital data out of the state elements
to capture logic levels generated at other nodes in
response to the test stimuli. Such control and
observation of an IC's electrical nodes via state elements
30 is also used to provide "visibility" into an IC for
debugging purposes. Further, an IC that supports scan

testing is frequently used to access electrical nodes of other electronic circuits connected thereto. For example, these other electronic circuits may be embedded within the IC, i.e., embedded memories or cores, or externally
5 connected to the IC.

Scan testing is typically performed in accordance with the IEEE 1149.1 Standard described in the IEEE 1149.1-1990 Standard Test Access Port and Boundary Scan Architecture specification. The IEEE 1149.1 Standard was
10 primarily developed to solve problems related to Printed Circuit Board (PCB) testing. The IEEE 1149.1 Standard is also typically used to access "scan chains" within ICs to facilitate testing and debugging of ICs, PCBs, and systems.

15 "Boundary scan" is an application of scan testing at input and output (I/O) pins of an IC to provide direct control and observation of electrical nodes using boundary scan operations. Boundary scan involves a specific type of scan path having a boundary scan register cell at each
20 I/O pin of an IC. For example, by performing boundary scan operations, known logic levels may be placed directly on outputs of one circuit and observed at inputs of another circuit connected thereto. Boundary scan therefore provides a way of determining whether circuits
25 are properly connected to each other on, e.g., a PCB, and/or whether there are manufacturing defects on the PCB that may prevent the circuits from carrying out their intended mission. Boundary scan tests can detect different types of defects on a PCB, e.g., broken circuit
30 traces, cold solder joints, solder bridges, and

electrostatic-discharge (ESD) induced failures in IC buffers.

Fig. 1 depicts boundary scan architecture 100, which is compliant with the IEEE 1149.1 Standard and may be embedded in an IC to provide direct control and observation of electrical nodes via the IC's I/O pins. As depicted in Fig. 1, an IC compliant with the IEEE 1149.1 Standard includes the following four (4) mandatory pins: TDI, TDO, TCK, and TMS. Further, the IC optionally includes the pin, TRSTN. These pins, TDI, TDO, TCK, TMS, and optionally TRSTN, are commonly known as the Test Access Port (TAP).

In addition, the IC includes three (3) mandatory scan registers, i.e., an Instruction Register (IR) 104; and, two (2) Data Registers (DRs), i.e., a Boundary Scan Register (BSR) 106 and a Bypass Register (BYPASS) 108. Further, the IC optionally includes at least one User DR, i.e., a User DR 110, which may be used to implement tests such as internal scan path testing and Built-In Self-Test (BIST). Moreover, the IC includes a protocol interface 102 known as a TAP Controller, which includes a 16-state Finite State Machine (FSM) operated by the mandatory TMS and TCK input pins and other logic.

Fig. 3 depicts a state diagram 300 for the standard FSM of the Tap Controller 102. A logic level on the TMS pin determines a next state of the FSM, and a clock signal on the TCK pin causes state transitions to occur. Further, an updated IR instruction selects the IR 104, the BSR 106, the BYPASS DR 108, or the USER DR 110 for scan operations. Moreover, the FSM includes a Select-DR branch 301 that defines states for performing a "DR-scan"

operation and a Select-IR branch 303 that defines states for performing an "IR-scan" operation.

The IR 104, the BSR DR 106, the BYPASS DR 108, and the USER DR 110 each comprise a separate scan path; and, 5 the TAP Controller 102 enables operation of only one of these scan registers at a time. The selected scan register shifts its scan data between the TDI pin and the TDO pin during a Shift-IR 324 state or a Shift-DR 310 state. Further, the selected scan register for the next 10 set of scan operations is determined by the IR instruction that was previously updated in the IR 104 during an Update-IR 332 state. Moreover, the IEEE 1149.1 Standard uses both edges of the clock signal on the TCK pin. Specifically, a logic level on the TMS pin and scan data 15 on the TDI pin are sampled on the rising edge of the clock signal, and scan data on the TDO pin changes on the falling edge of the clock signal.

The state diagram 300 of Fig. 3 includes six (6) "steady-states," i.e., a Test-Logic-Reset 302 state, a 20 Run-Test/Idle 304 state, a Shift-DR 310 state, a Pause-DR 314 state, a Shift-IR 324 state, and a Pause-IR 328 state. According to the IEEE 1149.1 Standard, there is only one (1) steady-state when the TMS pin is set to logical 1, i.e., the Test-Logic-Reset 302 state; and, the FSM of the 25 TAP Controller 102 can be reset (i.e., transition to the Test-Logic-Reset 302 state) within five (5) TCK clock signal transitions while the TMS pin is set to logical 1. The optional TRSTN pin provides another way to reset the FSM of the TAP Controller 102. For example, setting the 30 TRSTN pin to logical 0 causes an asynchronous reset of the TAP Controller 102 FSM.

The DR branch 301 and the IR branch 303 of the TAP Controller 102 FSM each includes six (6) states. Specifically, a Capture state, e.g., a Capture-DR 308 state and a Capture-IR 322 state, causes a selected scan register to capture data via parallel inputs of the scan register. This captured data is shifted out of the selected register on the TDO pin during the Shift-DR 310 state or the Shift-IR 324 state, while new scan data is simultaneously shifted into the register via the TDI pin.

After the scan shift operation is completed, new scan data is updated into a parallel update stage of the scan register. An IR or DR update operation is enabled upon entering either the Update-IR 332 state or the Update-DR state 318, respectively.

A state sequence including an Exit1-DR 312 state, immediately followed by a Pause-DR 314 state, and immediately followed by an Exit2-DR 316 state is used to terminate or suspend the Shift-DR 310 state operation. Similarly, a state sequence including an Exit1-IR 326 state, immediately followed by a Pause-IR 328 state, and immediately followed by an Exit2-IR 330 state is used to terminate or suspend the Shift-IR 324 state operation. The Pause-DR 314 state and the Pause-IR 328 state are included in the TAP Controller 102 FSM primarily to account for potentially slow tester hardware and/or software performance.

The instructions updated in the IR 104 not only select scan registers for shifting scan data, but also determine test behavior, e.g., for boundary scan testing of PCB interconnects. The IEEE 1149.1 Standard specifies three (3) mandatory instructions and their corresponding

test behavior. One such instruction is the BYPASS instruction, which selects the BYPASS DR 108 to provide a 1-bit scan path between TDI and TDO. For example, scan paths of multiple circuits compliant with the IEEE 1149.1 Standard may be chained together by serially connecting their respective TDI and TDO pins. The BYPASS instruction may then be used to "bypass" these potentially long scan paths when DRs corresponding thereto do not have to be accessed for particular test or debug operations. In this way, the BYPASS instruction(s) may be used to reduce the number of scan bits that are shifted.

Other mandatory instructions are the SAMPLE/PRELOAD and EXTEST instructions, both of which select the BSR 106. Specifically, the SAMPLE/PRELOAD instruction is used to sample logic levels on an IC's I/O pins during normal operation and pre-load data into the BSR 106 before testing. The IC's I/O pins are controlled by normal system logic during the SAMPLE/PRELOAD instruction. The EXTEST instruction controls the IC's I/O pins from the BSR 106 and may be used to perform PCB interconnect testing.

Although the boundary scan architecture 100 can be used to program, test, and debug electronic circuits, the boundary scan architecture 100 has several drawbacks. For example, in accordance with the IEEE 1149.1 Standard, the Pause-IR 328 state and the Pause-DR 314 state are included in the Select-IR branch 303 and the Select-DR branch 301, respectively, of the TAP Controller 102 FSM primarily to account for less than optimal tester hardware and/or software performance. However, the hardware and software of today's test resources are normally fast enough to shift scan data continuously without requiring the Pause-

IR 328 and Pause-DR 314 states. Accordingly, these states merely increase the time required to perform such shift operations.

In addition, in ICs compliant with the IEEE 1149.1 Standard and PCBs including such components, it is frequently difficult to access electrical nodes of non-scan sequential circuits from scan paths surrounding these circuits. Such non-scan sequential circuits may include non-scan embedded cores and component clusters on PCBs. The IEEE 1149.1 Standard is particularly inefficient when used to access memories, whether the memories are embedded within an IC or externally connected to an IC. This is because memories typically require long and/or complex control and address sequences to fully access their contents and provide comprehensive testing thereof, and ICs compliant with the IEEE 1149.1 Standard often cannot efficiently handle such long and complex sequences.

Specifically, if the BSR 106 of an IC were used to move data to a memory externally connected to the IC, then read/write cycles of the memory would typically be emulated using the Select-DR branch 301 of the TAP Controller 102 FSM, and data would typically be shifted into the BSR 106 to sequence the data, address, and control signals of the memory. This is conventionally done by putting the IC into the EXTEST mode, shifting a pattern of address, data, and control values into corresponding bit positions of the BSR 106, and updating the BSR 106 to apply the bit pattern to the memory in parallel.

However, any change in a control value, e.g., to place a logical 0 on a write enable (WEN) pin, therefore

requires multiple BSR shift operations. This can be very inefficient for complex access sequences, wherein several BSR scan operations may be required to read from or write to a single address location of the memory. Further, read/write control sequences must be serialized to apply these same sequences through the BSR 106.

Still further, each BSR scan operation scans all of the bits in the BSR 106, not just those bits needed to access the memory data, address, and control signals. This also decreases efficiency because in a typical IC with high-density I/O packaging, e.g., a Ball Grid Array (BGA), the BSR 106 can be several hundred bits in length yet only a fraction of these bits typically needs to be accessed to control the memory signals.

In addition, the protocol of the TAP Controller 102 FSM prohibits certain types of testing. Specifically, the TAP Controller 102 FSM does not provide for a state sequence including the Shift-DR state, immediately followed by the Capture-DR state, and immediately followed by the Shift-DR state, which is normally required for controlling internal scan registers; or, a state sequence including the Update-DR state immediately followed by the Capture-DR state, which is normally required for performing path delay testing of interconnects.

It would therefore be desirable to have improved apparatus and methods that provide for the access of electrical nodes of electronic circuits including scan and non-scan sequential circuits and combinatorial circuits. It would also be desirable to have improved apparatus and methods that efficiently access electrical nodes of

electronic circuits that require long and/or complex control and address sequences.

BRIEF SUMMARY OF THE INVENTION

5 In accordance with the present invention, apparatus and methods are disclosed for providing optimized access to electrical nodes of electronic circuits for programming, testing, and debugging. The presently disclosed apparatus includes an access interface circuit that can be connected
10 between a test resource and an electronic circuit such as a circuit adapted for scan testing, a non-scan sequential circuit, a component cluster, or a memory circuit. The access interface circuit includes a protocol generator, which has a control register queue and an address register;
15 and, a data generator, which has a plurality of registers adapted for capturing, shifting, and/or updating data samples. Address and control sequences for programming/testing/debugging the electronic circuit can be serially loaded into the control register queue and the
20 address register using the test resource. Data sequences for programming/testing/debugging the electronic circuit can also be serially loaded into the data registers using the test resource. The access interface circuit also includes circuitry for switchably and directly applying either the
25 serially loaded address and control sequences or the normal system address and control signals to the electronic circuit; and, for switchably and directly applying either the serially loaded data sequences or the normal system data signals to the electronic circuit. The access interface
30 circuit further includes circuitry for directly capturing data values from the electronic circuit in the data

registers. The access interface circuit therefore allows complex access sequences to be efficiently applied to the electronic circuit, and data sequences to be efficiently written to and read from the electronic circuit under control of the test resource. Such capability is particularly useful in programming/testing/debugging memory and non-scan circuits.

The presently disclosed apparatus further includes a programmable controller connected to the access interface circuit and connectable to the test resource. The programmable controller includes a state machine having a plurality of states at least one of which is programmable. The access interface circuit performs operational commands based on at least a portion of the plurality of states. Because the state machine is programmable, the programmable controller enhances the flexibility of the access interface circuit for programming/testing/debugging electronic circuits. The access interface circuit also includes at least one scan path connectable to the electronic circuit; and, circuitry for switchably applying, under control of the programmable controller, address/control sequences from the protocol generator, data sequences from the data generator, and/or data sequences from the test resource to the scan path. The control register queue and address register of the protocol generator and the data registers of the data generator comprise minimal length scan paths that can be used for efficiently programming/testing/debugging the circuits adapted for scan testing.

The access interface circuit and/or the programmable controller may be controlled by way of the test resource to enable communication between the test resource and the

electronic circuit connected to the access interface circuit. The test resource can be any type of apparatus customarily used to program/test/debug electronic circuits, such as Automatic Test Equipment (ATE) or a general-purpose computer, e.g., a Personal Computer (PC), connected to the
5 access interface circuit. Alternatively, the test resource may be embedded with the access interface circuit and the programmable controller in a device such as an IC.

The access interface circuit and/or the programmable
10 controller may be designed into a device as a permanent/fixed interface; or, in the case of a programmable logic device such as a Field Programmable Gate Array (FPGA), the access interface/programmable controller may be temporarily programmed into the device. By utilizing the
15 latter method, electronic circuits coupled to the access interface may be temporarily accessed for programming/testing/debugging purposes via the FPGA, and the FPGA can be subsequently re-programmed with its normal mission mode logic.

20 Other features, functions, and aspects of the access interface will be evident from the Detailed Description of the Invention that follows.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

25 The invention will be more fully understood by reference to the following Detailed Description of the Invention in conjunction with the Drawing of which:

Fig. 1 is a block diagram depicting the Standard IEEE 1149.1 Test Access Port and Boundary Scan Architecture;

Fig. 2 is a block diagram depicting a test configuration operative in a manner according to the present invention;

Fig. 3 is a flow diagram depicting the operation of the finite state machine for the Standard IEEE 1149.1 Test Access Port Controller;

Fig. 4 is a block diagram depicting an access interface and a controller used in the test configuration of Fig. 2;

Fig. 5 is a flow diagram depicting the operation of the finite state machine for the controller of Fig. 4;

Fig. 6 is a block diagram depicting a data generator included in the access interface of Fig. 4;

Fig. 7 is a block diagram depicting an alternative embodiment of the data generator of Fig. 6;

Fig. 8 is a block diagram depicting a protocol generator included in the access interface of Fig. 4;

Fig. 9 is a plan view of a protocol register included in the protocol generator of Fig. 8;

Fig. 10 is a timing diagram depicting a first operational sequence of the access interface and the controller of Fig. 4;

Fig. 11 is a timing diagram depicting a second operational sequence of the access interface and the controller of Fig. 4;

Fig. 12 is a block diagram depicting the access interface and the controller of Fig. 4 configured to access electrical nodes of parallel memory circuits;

Fig. 13 is a block diagram depicting the access interface and the controller of Fig. 4 configured to access electrical nodes of serial memory circuits;

Fig. 14 is a flow diagram depicting a first exemplary method of operation for the access interface and the controller of Fig. 4; and

Fig. 15 is a flow diagram depicting a second exemplary method of operation for the access interface and the controller of Fig. 4.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 2 depicts an illustrative embodiment of a test configuration 200 operative in a manner according to the present invention. Specifically, the test configuration 200 includes an access interface 206 that allows communications between a test resource 204 and electronic circuits 208 and 210 under test. In this illustrative embodiment, both the access interface 206 and the electronic circuit 208 are embedded in an IC 202; and, the electronic circuit 210 external to the IC 202 is connected to the access interface 206 and the circuit 208 by way of a bus 214 coupled to at least one I/O pin of the IC 202. Similarly, the test resource 204 is connected to the access interface 206 by way of a bus 212 coupled to at least one I/O pin of the IC 202.

In one embodiment, the IC 202 used to implement the access interface 206 may be an Application Specific Integrated Circuit (ASIC). In this case, the access interface 206 has a fixed configuration, although it may be configured to access any number or types of circuits such as electronic circuits 208 or 210. In another embodiment, the IC 202 used to implement the access interface 206 may be a programmable device such as an FPGA. In this case, the FPGA may be programmed to implement the access interface 206 via

the test resource 204, and different configurations of the access interface 206 may be re-programmed into in the FPGA to access any number or types of circuits. After testing, debugging, or programming devices coupled to the access
5 interface 206, the FPGA may be re-programmed to perform its intended system function.

Those of ordinary skill in this art will appreciate that other test configurations are possible. For example, the access interface 206 may be implemented on an IC that
10 is separate from any IC including circuits under test. Further, the test resource 204 may alternatively comprise a programming or debugging resource, and the combination of the test resource 204 and the access interface 206 may be used to program or debug the circuits 208 and 210.

15 The test resource 204 may comprise at least one memory such as a ROM or RAM, and at least one processor operative for executing programs stored in the memory, including applications for communicating with the access interface 206 to program, test, or debug the circuits 208
20 and 210. Further, as depicted in Fig. 2, the test resource 204 may be externally connected to the IC 202 incorporating the access interface 206. Alternatively, the test resource 204 may be embedded in the IC 202.

It should be understood that the test configuration
25 200 may include more than one access interface; and, each access interface may be connected to one or more electronic circuits such as the embedded circuit 208 and the external circuit 210.

Fig. 4 depicts an illustrative embodiment of the
30 access interface 206 that can be used to program, test, or debug the circuits 208 and 210. Specifically, the access

interface 206 is connected to a protocol interface 402 and an Instruction Register (IR) 404. In one embodiment, the protocol interface 402 comprises the standard TAP Controller 102 (see Figs. 1 and 3). In a preferred
5 embodiment, the protocol interface 402 is a modified TAP Controller, which will be described in detail below. In this illustrative embodiment, the access interface 206, the modified TAP Controller 402, the IR 404, and multiplexors 406 and 408 connected thereto, are embedded
10 in the IC 202 (see Fig. 2) and operatively connected to the test resource 204 and the circuits 208 and 210.

More specifically, the modified TAP Controller 402 is connected to the test resource 204 by way of the bus 212, which is coupled to input pins TRSTN, TMS, and TCK of the
15 IC 202. Similarly, the access interface 206, the IR 404, and the multiplexor 406 are connected to the test resource 204 by way of the bus 212, which is also coupled to an input pin TDI and an output pin TDO. Still further, the access interface 206 is connected to the circuits 208 and
20 210 by way of the bus 214, which is coupled to a CONTROL bus 418, an ADDRESS bus 419, and a bi-directional DATA bus 420.

In this illustrative embodiment, the TDI, TDO, TCK, and TMS pins correspond to the mandatory pins of the IEEE
25 1149.1 Standard; and, the TRSTN pin corresponds to the optional input pin of the IEEE 1149.1 Standard. Accordingly, registers included in the access interface 206 can serially receive data via the TDI pin, and serially transmit data on the TDO pin by way of the
30 multiplexors 406 and 408.

In addition, the modified TAP Controller 402 includes conventional circuitry for providing an AI_RESET signal on a line 430 and DR_CNTL signals on a bus 432 to the access interface 206; and, the IR 404 includes conventional
5 circuitry for providing DR_SEL signals on a bus 434 and a AI_ENABLE signal on a line 436 to the access interface 206. Specifically, the DR_SEL signals are used to select a register in the access interface 206; the DR_CNTL signals are used to control register operations (e.g.,
10 Capture, Shift, and Update operations) using the selected register; the AI_RESET signal is used to reset the access interface 206; and, the AI_ENABLE signal is used to enable the access interface 206. The operation of these signals will be described in detail below.

15 As depicted in Fig. 4, the access interface 206 includes a protocol generator 410, which is connected to a multiplexor 414 for providing control signals in programmed sequences to the circuits 208 and 210 on the bus 418. Specifically, the multiplexor 414 receives a
20 signal from the protocol generator 410 on lines 428 to select between normal system control signals of the circuits 208 and 210 on a bus 424 and programmed protocol signals on a bus 422. In this illustrative embodiment, the normal system control signals are provided by the IC
25 202 (see Fig. 2).

Similarly, the protocol generator 410 is connected to a multiplexor 415 and can be used to provide address signals in programmed sequences to the circuits 208 and 210 on the bus 419. Specifically, the multiplexor 415
30 receives a signal from the protocol generator 410 on the lines 428 to select between normal system address signals

of the circuits 208 and 210 on a bus 425 and programmed address signals on a bus 423. In this illustrative embodiment, the normal system address signals are provided by the IC 202 (see Fig. 2). The protocol generator 410
5 further includes a sequence generator 802 comprising a plurality of registers, e.g., registers SR1 through SRN, and an address register (AR) 804 (see Fig. 8), that are connected between the TDI and TDO pins.

The access interface 206 also includes a data
10 generator 412, which includes another plurality of registers, e.g., a register 601 comprising registers 602 and 604 (see Fig. 6), and separate registers 702 and 704 (see Fig. 7), that are connected between the TDI and TDO pins. Specifically, the registers of the data generator
15 412 receive the DR_SEL signals from the IR 404 and the DR_CNTL signals from the modified TAP Controller 402 for use in performing, e.g., Capture, Shift, and Update operations.

The data generator 412 is connected to a multiplexor
20 416 for providing data signals to the circuits 208 and 210 on the bus 420. Specifically, the multiplexor 416 receives a signal from the protocol generator 410 on the lines 428 to select between the data signals generated by the data generator 412 on bus 426 and normal system data
25 signals of the circuits 208 and 210 on bus 429. In this illustrative embodiment, the normal system data signals are provided by the IC 202 (see Fig. 2).

The registers of the data generator 412 vary in length depending upon specific applications and numbers of
30 data inputs and outputs of the circuits 208 and 210. In a preferred embodiment, the registers of the data generator

412 include as many cells or bits as there are data inputs and outputs of the circuits 208 and 210.

The registers of the data generator 412 also provide specialized data access and/or test functions. For example, in BIST implementations, the data generator 412 may include a Linear Feedback Shift Register (LFSR) for providing test stimuli and a Multiple Input Signature Register (MISR) for capturing a circuit response.

Fig. 5 depicts a state diagram 500 for the FSM of the modified TAP Controller 402. Specifically, the state diagram 500 includes eight (8) steady-states, i.e., a Test-Logic-Reset 502 state, a Run-Test/Idle 504 state, a Shift-DR 510 state, a Pause-DR 514 state, a Shift-IR 524 state, a Pause-IR 528 state, an Access1-DR 536 state, and an Access2-DR 540 state. Further, the state diagram 500 includes a Select-IR branch 503 that defines a sequence of states for performing an "IR-scan" operation, a Select-DR branch 501 that defines a sequence of states for performing "DR-scan" operations, and an access interface branch 505 that defines a sequence of states for performing programmable access operations using the access interface 206.

The logic level of the AI_ENABLE signal (initially logical 0; "AI_ENABLE is low") and the logic level at the TMS pin determine a next state of the modified TAP Controller 402 FSM, and the clock signal at the TCK pin ("the TCK clock") causes state transitions to occur. Specifically, the state diagram 500 shows that the Test-Logic-Reset 502 state is a steady-state when the level at the TMS pin is logical 1 ("TMS is high"). Next, if the level at the TMS pin changes from logical 1 to logical 0

("TMS is low"), then there is a transition from the Test-Logic-Reset 502 state to the Run-Test/Idle 504 state at the next rising edge of the TCK clock. The Run-Test/Idle 504 state is a steady-state when TMS is low.

5 If the level at the TMS pin now changes from low to high, then there is a transition from the Run-Test/Idle 504 state to the Select-DR-Scan 506 state at the next rising edge of the TCK clock. As depicted in Fig. 5, either the Select-DR branch 501 or the access interface
10 branch 505 can be selected as the next sequence of states from the Select-DR-Scan 506 state. Otherwise, at the next rising edge of the TCK clock, there is a transition from the Select-DR-Scan 506 state to the Select-IR-Scan 520 state. As depicted in Fig. 5, the Select-IR branch 503
15 can be selected as the next sequence of states from the Select-IR-Scan 520 state.

 Specifically, if the level of the TMS pin changes from high to low when in the Select-DR-Scan 506 state, then at the next rising edge of the TCK clock there is a
20 transition from the Select-DR-Scan 506 state to the Capture-DR 508 state, which is the first state in the Select-DR branch 501. Similarly, if the level of the TMS pin changes from high to low when in the Select-IR-Scan 520 state, then at the next rising edge of the TCK clock
25 there is a transition from the Select-IR-Scan 520 state to the Capture-IR 522 state, which is the first state in the Select-IR branch 503. In this illustrative embodiment, the state sequences in the Select-DR branch 501 (i.e., states 508, 510, 512, 514, 516, and 518) and the Select-IR
30 branch 503 (i.e., states 522, 524, 526, 528, 530, and 532) are the same as those in the corresponding Select-DR

branch 301 and Select-IR branch 303, respectively, of the standard TAP Controller 102 FSM (see Figs. 1 and 3).

Further, if the level of the TMS pin changes from high to low and the level of the AI_ENABLE signal changes from low to high ("AI_ENABLE is high") when in the Select-DR-Scan 506 state, then at the next rising edge of the TCK clock there is a transition from the Select-DR-Scan 506 state to an Enter-Access-DR 534 state, which is the first state in the access interface branch 505. As depicted in Fig. 5, each branch 501, 503, and 505 of the state diagram 500 is compliant with the IEEE 1149.1 Standard in that the modified TAP Controller 402 FSM can be reset to the Test-Logic-Reset 502 state within 5 cycles of the TCK clock when TMS is high.

As mentioned above, in this illustrative embodiment, the Select-DR branch 501 and the Select-IR branch 503 of the state diagram 500 correspond to the Select-DR branch 301 and the Select-IR branch 303, respectively, of the standard TAP Controller 102 FSM. Accordingly, the operations performed in the respective state sequences of the Select-DR branch 501 and the Select-IR branch 503 are fixed. In contrast, the operations performed in the state sequence of the access interface branch 505 (i.e., states 534, 536, 538, 540, 542, and 544) are programmable.

For example, the operations performed in the state sequence of the access interface branch 505 may be programmed in response to a predetermined event such as the provision of an electronic circuit with internal scan registers. Accordingly, the access interface branch 505 may be programmed to generate a repeatable Shift/Capture

sequence, which may then be used to control operations using the internal scan registers.

Specifically, if the level of the TMS pin remains low when in the Enter-Access-DR 534 state, then at the next rising edge of the TCK clock there is a transition from the Enter-Access-DR 534 state to the Access1-DR 536 state. For example, the Access1-DR 536 state may be programmed to perform either a Shift operation on a selected register or a Pause operation. Further, the Access1-DR 536 state is a steady-state when TMS is low.

If the level of the TMS pin changes from low to high when in the Access1-DR 536 state, then at the next rising edge of the TCK clock there is a transition from the Access1-DR 536 state to an Updatel/Capture1-DR 538 state. For example, the Updatel/Capture1-DR 538 state may be programmed to perform an Update operation only, a Capture operation only, or both the Update and Capture operations. Specifically, when the Updatel/Capture1-DR 538 state is programmed to perform both the Update and Capture operations, at the next falling edge of the TCK clock after transitioning to the Updatel/Capture1-DR 538 state, the Update operation is performed; and, at the next rising edge of the TCK clock, the Capture operation is performed followed by a transition from the Updatel/Capture1-DR 538 state to the next state.

If the level of the TMS pin now changes from high to low, then at the next rising edge of the TCK clock there is a transition from the Updatel/Capture1-DR 538 state to the Access2-DR 540 state. Like the Access1-DR 536 state, the Access2-DR 540 state may be programmed to perform either a Shift operation on the selected register or a

Pause operation. Further, the Access2-DR 540 state is a steady-state when TMS is low.

If the level of the TMS pin changes from low to high when in the Access2-DR 540 state, then at the next rising edge of the TCK clock there is a transition from the Access2-DR 540 state to an Update2/Capture2-DR 542 state. Like the Updatel/Capturel-DR 538 state, the Update2/Capture2-DR 542 state may be programmed to perform an Update operation only, a Capture operation only, or both the Update and Capture operations.

Finally, if the level of the TMS pin remains high, then at the next rising edge of the TCK clock there is a transition from the Update2/Capture2-DR 542 state to an Exit-Access-DR 544 state, which is the last state in the access interface branch 505.

It should be noted that if the level of the TMS pin changes from low to high when in the Enter-Access-DR 534 state, then at the next rising edge of the TCK clock there is a transition from the Enter-Access-DR 534 state to the Updatel/Capturel-DR 538 state. Further, if the level of the TMS pin remains high when in the Updatel/Capturel-DR 538 state, then at the next rising edge of the TCK clock there is a transition from the Updatel/Capturel-DR 538 state to the Exit-Access-DR 544 state. Still further, if the level of the TMS pin changes from high to low when in the Update2/Capture2-DR 542 state, then at the next rising edge of the TCK clock there is a transition from the Update2/Capture2-DR 542 state to the Access1-DR 536 state.

When AI_ENABLE is high, the access interface 206 is enabled. As a result, subsequent operations are performed on selected registers included in the access interface

206. It should be understood that instructions loaded into the IR 404 not only select the registers in the access interface 206, but also program the states 536, 538, 540, and 542 of the access interface branch 505. The operations are then sequenced as the modified TAP Controller 402 FSM follows the access interface branch 505. The access interface 206 may have any number of instructions corresponding thereto, depending upon the circuits to be accessed and the desired applications.

10 Specifically, if the instruction loaded into the IR 404 selects a register of the data generator 412 and programs the Access1-DR 536 state to perform a Shift operation, then upon entering the Access1-DR 536 state, data in the selected register may be shifted while the protocol generator 410 concurrently executes a sequence of access operations on the circuit 208 or 210. Further, the amount of data shifted and the access sequence length may be precisely controlled based upon the number of TCK clock cycles that occurs when in the Access1-DR 536 state.

15 Alternatively, the instruction loaded into the IR 404 may program the Access1-DR 536 state to perform a Pause operation only to pause the data generator 412 for a predetermined number of TCK clock cycles while the protocol generator 410 concurrently executes. The

20 Access2-DR 540 state may be programmed in a similar manner.

The Update1/Capture1-DR 538 and Update2/Capture2-DR 542 states also allow programmable Update and Capture operation sequences to occur based upon instructions loaded into the IR 404. For example, the instruction loaded into the IR 404 may select a register of the data

30

-24-

generator 412, program the Updatel/Capture1-DR 538 state to perform an Update operation, and program the Update2/Capture2-DR 542 state to perform a Capture operation using the selected register.

5 In addition, the Access1-DR 536 state, the Updatel/Capture1-DR 538 state, and the Access2-DR 540 state may be similarly programmed to generate a Shift/Capture/Shift sequence, which may be used to control operations on internal scan registers. In this way, an
10 Update operation, which is typically not required for controlling operations on internal scan registers, can be eliminated from this sequence.

 In addition, both the Updatel/Capture1-DR 538 and Update2/Capture2-DR 542 states may be programmed to
15 generate an Update operation immediately followed by a Capture operation. This Update/Capture sequence makes it possible to perform path delay testing of interconnects using standard boundary scan cells.

 In addition, the Updatel/Capture1-DR 538 and Access2-
20 DR 540 states may be programmed to generate a Capture/Shift sequence, which may be used to capture read data from a memory circuit into a register of the data generator 412 and then immediately start shifting the read data out.

25 In addition, either the Updatel/Capture1-DR 538 state or the Update2/Capture2-DR 542 state may be programmed to inhibit both the Update and the Capture operations. As a result, these states may be used for merely starting, stopping or continuing execution of the protocol generator
30 410.

Upon entering the Exit-Access-DR 544 state, the access interface 206 is disabled. Further, depending upon whether TMS is high or low, at the next rising edge of the TCK clock there may be a transition from the Exit-Access-DR 544 state to either the Run-Test/Idle 504 state to "park" the modified TAP Controller 402 or to the Select-DR-Scan 506 state to start a new operation.

Figs. 6 and 7 depict block diagrams of illustrative embodiments of the data generator 412. As depicted in Fig. 6, the data generator 412 includes the register 601 comprising the register 602 for performing Shift/Capture operations and the register 604 for performing Update operations. In a preferred embodiment, both the registers 602 and 604 include as many cells as there are data inputs and outputs in, e.g., the circuits 208 and 210. For example, the data generator 412 may include a plurality of registers such as the register 601.

Instructions loaded into the IR 404 are used to generate (1) the DR_SEL signals for selecting a register in the data generator 412, e.g., the register 601, and (2) the AI_ENABLE signal for enabling operation of the access interface 206 including the data generator 412. Further, the modified TAP Controller 402 generates (1) the AI_RESET signal for resetting the access interface 206, and (2) the DR_CNTL signals for controlling the Shift/Capture/Update operations of the selected register. It should be understood that the manner in which the registers of the data generator 412 are enabled, selected, re-set, and controlled is conventional.

In addition, the register 602 shifts-in data at the TDI pin and shifts-out data on the TDO pin. Further, the

timing of the Shift/Capture/Update operations performed by the register 601 are synchronized with either the TCK clock or with a system clock of the circuit 208 or 210.

Specifically, the register 601 can be used for
5 accessing data portions of the circuits 208 and 210 using simple register operations. For example, the register 602 may capture data from the circuits 208 and 210, and the register 604 may update data to the circuits 208 and 210. The Capture operation may be performed following an access
10 sequence that causes the circuit 208 or 210 to output data onto parallel inputs 606 of the register 602. That data can then be latched into the register 602 during, e.g., the Update1/Capture1-DR 538 state of the access interface branch 505 (see Fig. 5). Next, that data can be serially
15 shifted-out to the TDO pin during, e.g., the Access2-DR 540 state, so that it can be observed. Further, the Update operation may be performed by first serially shifting data into the register 602 during, e.g., the Access1-DR 536 state, thereby loading the register 602.
20 When the Shift operation has completed, that data can be moved in parallel to the register 604 during, e.g., the Update1/Capture1-DR 538 state, for subsequent output to the circuit 208 or 210 via parallel outputs 610.

As depicted in Fig. 7, the data generator 412
25 alternatively includes at least two (2) separate registers connectable to the circuits 208 and 210, e.g., the register 702 for receiving data from the circuits 208 and 210 on parallel inputs 706 and outputting data to the circuits 208 and 210 on parallel outputs 708, and the
30 register 704 for receiving data from the circuits 208 and 210 on parallel inputs 710 and for outputting data to the

circuits 208 and 210 on parallel outputs 712. Specifically, the two (2) registers 702 and 704 are on separate scan paths. The register 702 is scanned using a first pair of TDI_1-TDO_1 pins; and, the register 704 is scanned using a second pair of TDI_2-TDO_2 pins. It should be understood that the manner in which the registers of the data generator 412 are enabled using the AI_ENABLE signal, selected using the DR_SEL signals, reset using the AI_RESET signal, and controlled using the DR_CNTL signals is conventional.

Fig. 8 shows a block diagram of an illustrative embodiment of the protocol generator 410. Specifically, the protocol generator 410 includes the sequence generator 802 and the address register 804. The sequence generator 802 provides a programmable way of applying a particular access protocol to the circuit 208 or 210; and, the address register 804 provides a programmable way of applying a sequence of addresses to the circuit 208 or 210.

The sequence register 802 includes a scannable control queue comprising the registers SR1 through SRN, which can apply logic levels directly (i.e., in parallel) to control inputs of the circuit 208 or 210 via the bus 422. Alternatively, the control queue contents may control or be decoded by additional logic within the protocol generator 410 to provide control signals to the circuit 208 or 210. For example, the additional logic may include a clock generator 810, which outputs a CLOCK signal on the bus 422. The clock generator 810 may be used, e.g., to provide an access clock such as the TCK clock or a system clock to the connected circuit 208 or

210. Accordingly, the clock generator 810 may include divider circuitry for setting a specific clock rate, or a pulse-width shaping circuit for accessing asynchronous memories using a clock that is synchronous to the access interface 206. During operation of the access interface 206, this additional logic may synchronize with the TCK clock or the system clock to allow it to operate at the system speed of the circuit 208 or 210.

In this illustrative embodiment, the control queue of the sequence register 802 is N-registers deep by M-bits wide. For example, the N-register deep control queue may be configured to support the maximum length protocol sequence to be applied to the circuit 208 or 210. These N control queue locations are depicted in Fig. 8 as SR1 through SRN.

Specifically, during operation of the access interface 206 (i.e., while AI_ENABLE is high), the N control queue locations are successively accessed and their contents used to sequence the control signals of the circuit 208 or 210 via the bus 422, and to control sequencing of the data generator 412 using DG_CNTL signals on a bus 488 (see also Fig. 4). Accordingly, each control queue location SR1 through SRN holds values for a single control state of the circuit 208 or 210 along with corresponding control states of the data generator 412.

In addition, when operation of the access interface 206 is disabled (i.e., AI_ENABLE is low), the contents of the registers SR1 through SRN can be scanned. Accordingly, the sequence register 802 can be programmed to set initial contents of the registers SR1 through SRN before enabling the access interface 206.

The programmability of the protocol generator 410 permits (1) the sequence length of the access protocol to vary, e.g., from 1 control sequence up to the maximum queue depth of N, and (2) the access protocol to be
5 repetitively applied without re-scanning the contents of the registers SR1 through SRN. Consequently, the sequence register 802 can apply various sequences of control signals in accordance with required circuit access protocols.

10 As mentioned above, the control queue locations SR1 through SRN also contain levels for controlling the access interface 206. As depicted in Fig. 8, the sequence register 802 provides AR_CNTL signals to the address register 804 via a bus 832, which may be used to select or
15 control a particular address sequence generated by the address register 804. The sequence register 802 also provides the DG_CNTL signals to the data generator 412 on the bus 830, which may be similarly used to select or control a particular data sequence generated by the data
20 generator 412.

The execution of the sequence register 802 is controlled by SR select logic 806 (see Fig. 8), which provides control signals to the sequence register 802. For example, the SR select logic 806 may be programmed via
25 scan initialization to vary the control queue depth from 1 to N. In a preferred embodiment, when the control queue depth is programmed to be less than N, upper control queue locations are not included in the access protocol sequence. The execution of the sequence register 802
30 starts based upon an instruction loaded in the IR 404, which causes the DR_SEL and AI_ENABLE signals to be

-30-

appropriately asserted. During the execution of the sequence register 802, the SR Select Logic 806 may either stop sequencing the sequence register 802 after the programmed number of locations have been executed, or
5 repeat that sequence until stopped by the modified TAP Controller 402.

As mentioned above, the address register 804 provides address sequences to the circuit 208 or 210 via the bus 423. For example, if the circuit 208 or 210 includes a
10 memory circuit, then the address register 804 may provide address sequences directly (i.e., in parallel) to the memory circuit instead of providing the address sequences via a scan register. As a result, the protocol generator 410 can access memories without having to perform a scan
15 operation each time a new memory location is to be accessed.

The address register 804 includes one or more address generation circuits. For example, these circuits may include up/down counters or pseudo-random sequence
20 generators such as an LFSR. Further, each address generation circuit may comprise a separate address register within the address register 804. The address register 804 also includes conventional control circuitry for the address generation circuits, e.g., to select an up
25 or down count sequence. The address register 804 can therefore be configured to support different access sequences for different types of memories. For example, the address register 804 may provide multiple address sequences to support the programming, testing, and
30 debugging of multi-ported memories.

As also mentioned above, the address register 804 receives control inputs from the sequence register 802 via the AR_CNTL signals. For example, the AR_CNTL signals may come directly from the sequence register 802 queue or may
5 alternatively be decoded by the address register 804. Further, the AR_CNTL signals may be used to direct one of the address generation circuits to advance to a next memory address. The address register 804 can also control the SR Select Logic 806 via PG_CNTL signals on a bus 834
10 and control the data generator 412 via the DG_CNTL signals on the bus 488, thereby allowing values or conditional events included in address sequences to control both protocol and data generation in the access interface 206.

In this illustrative embodiment, the sequence
15 register 802 is scannable; and, the address register 804 and the registers SR1 through SRN are included in the same scan path, which is connected between the TDI and the TDO pins. Accordingly, the address register 804 can be scanned along with the sequence register 802. Scanning
20 the address register 804 allows its address register contents and any control logic included therein to be initialized before enabling the access interface 206.

The length of the scan chain including the registers SR1 through SRN and address register 804 depends upon
25 specific configurations of the address register 804 and the control queue of the sequence register 802. In this exemplary embodiment, the sequence register 802 has a length of $(N \times M)$ bits, and the address register 804 has a length totaling all of the lengths of the address
30 registers included therein.

In an alternate embodiment, the sequence register 802 and the address register 804 are partitioned so as to be on more than one scan path within the protocol generator 410. For example, the (N x M) bits of the sequence register 802 may be on one scan path, and each register of the address register 804 may be on separate scan paths. In this alternate embodiment, the protocol generator 410 may have multiple pairs of TDI-TDO pins. It should be understood that the manner in which the registers SR1 through SRN and the address register 804 are enabled using the AI_ENABLE signal, selected using the DR_SEL signals, re-set using the AI_RESET signal, and controlled using the DR_CNTL signals is conventional.

Fig. 9 depicts an illustrative embodiment of a control queue location 900 included in the sequence register 802. The queue location 900 indicates various control field/bit assignments that are representative of the control required to program, test, or debug, e.g., an asynchronous SRAM device on a PCB or embedded in an IC.

The three most significant bits (MSBs), i.e., bits 3 through 5, are used to control logic within the protocol generator 410. Specifically, MSB 5 (SR_EOS) is used to indicate that this SR location is the end of the protocol sequence by signaling the SR Select Logic 806 to stop the control queue sequencing at this point. Further, bit 4 (AR_Next) is used to direct the address register 804 to advance to a next memory address; and, bit 3 (DG_Next) is used to direct the data generator 412 to generate a next data pattern, e.g., a foreground or background data pattern in the case of a memory BIST application.

As depicted in Fig. 9, the three least significant bits (LSBs) of the queue location 900 are control signals that can be directly applied to the circuit 208 or 210. For example, the circuit 208 or 210 may comprise an SRAM. Accordingly, bit 2 (CEN) is a chip enable signal, bit 1 (OEN) is an output enable signal, and LSB 0 (WEN) is a write enable signal for the exemplary SRAM device. By setting values for CEN, OEN and WEN in respective control queue locations, the access sequence for the SRAM can be programmed to perform, e.g., a read operation, a write operation, a read followed by a write operation, etc.

Fig. 9 depicts the TDI and TDO pin connections relative to the single queue location 900. In this illustrative embodiment, these TDI and TDO pin connections are chained from one queue location to the next for the entire sequence register 802 to place the registers SR1 through SRN on the same scan path.

In one embodiment, the queue location 900 may be shared and used to access other circuits in addition to the SRAM mentioned above. Because the sequence register 802 is programmable, the control queue bits can be redefined to accommodate requirements of the multiple circuits. Further, additional control queue bits may be defined if needed, e.g., when certain control bits are used with one circuit and unused for another circuit.

Accordingly, the protocol generator 802 and the data generator 804 can be configured to support any circuit coupled to the access interface 206. For example, the access interface 206 may be connected and dedicated to a single circuit, or it may be implemented such that it is shared to provide access to a plurality of connected

circuits such as the circuits 208 and 210. The access interface 206 may also be partitioned among a plurality of circuits and/or devices. For example, the data generator 412 may be shared between a group of memories while using
5 separate protocol generators 802, each one dedicated for use with a respective memory. This would allow, e.g., a different BIST algorithm for each memory to be programmed in corresponding protocol generators, and the different BIST algorithms to be executed concurrently using the
10 access interface 206.

Alternatively, a plurality of access interfaces may operate in parallel from a single TAP Controller 402, or a single access interface may be partitioned across multiple ICs and synchronized to run from multiple TAP controllers
15 402. For example, this configuration may be used when an external memory structure is connected to multiple ASICs on a PCB, with a first ASIC controlling a memory data bus and a second ASIC providing address and control signals to the memories. In this application, the access interface
20 206 may be partitioned so that the data generator 412 is implemented in the first ASIC and the protocol generator 410 is implemented in the second ASIC. Each ASIC would then have a TAP Controller that would be used to run the access interface 206 partitioned across the two ASICs.

25 The embodiments disclosed herein will be better understood with reference to the following illustrative examples. In a first illustrative example, a memory circuit having 8-bits of input data and 8-bits of output data is connected to the access interface 206, which
30 accesses this memory circuit using the data generator 412, as depicted in Fig. 6. Specifically, the data generator

412 is implemented as dedicated logic using eight (8) BSR-type cells, and therefore has a scan length of 8-bits. In this first illustrative example, a Capture operation is used during memory read operations to capture data read from the memory into the data generator 412 on the inputs 5 606. The test resource 204 (see Fig. 2) then scans the 8-bits of captured data, which is read out of the data generator 412 on the TDO pin of the device, to observe the accessed memory data. Similarly, memory write data may be scanned into the data generator 412; and, following an 10 Update operation, a programmed access sequence of the protocol generator 410 may write this byte of data to the memory on the outputs 610.

In a second illustrative example, the access 15 interface 206 of Fig. 12 is used to access an external SRAM memory device. For example, the circuitry depicted in Fig. 12 may be implemented inside an ASIC or programmed in an FPGA, and that device may be connected to the external SRAM on a PCB.

20 The access interface 206 of Fig. 12 includes the sequence register 802, the address register 804, and the data generator 412. In this second illustrative example, the multiplexors 414, 415, and 416 that select between the normal system signals and the signals generated by the access interface 206 are not explicitly shown. 25 Specifically, the multiplexors 414, 415, and 416 are implemented with logic external to the access interface 206 but inside the same device containing the TAP Controller 402 and the access interface 206. The access 30 interface 206 provides a signal on the lines 428 to select between the signals at the inputs of the multiplexors 414,

415, and 416. The access interface 206 also provides ADDRESS signals on the bus 419, and a set of CONTROL signals, i.e., CEN, OEN and WEN, on the bus 418. These control signals are the same as those indicated in bits 0 through 2 of the queue location 900 (see Fig. 9). Data passing between the external SRAM and the data generator 412 on the bus 420 are also depicted in Fig. 12. All of the signals transmitted to and received from the memory are selected at the multiplexors 414, 415, and 416. These signals will be selected and connected to corresponding data, address, and control terminals of the SRAM during operation of the access interface 206.

In a third illustrative example, the access interface 206 of Fig. 13 is used to access a serial EEPROM device. The circuitry depicted in Fig. 13 may also be implemented inside an ASIC or programmed in an FPGA, and that device may be connected to the serial EEPROM device on a PCB.

The TAP Controller 402 of Fig. 13 is identical to that of Fig. 12. However, in this example the scan logic has been adapted for accessing the serial EEPROM device. Further, the access interface 206 of Fig. 13 shows only the sequence register 802 and the clock generator 810. The address register 804 and the data generator 412 are not explicitly shown because in this example they are not required for accessing the serial EEPROM device.

As in the SRAM example relating to Fig. 12, the access interface 206 generates control signals, i.e., CSN, HOLDN, and SCK, on the bus 418 to the serial EEPROM device. These signals enable chip operations and provide for clocking of the serial EEPROM device. The access interface 206 of Fig. 13 also provides select signals on

the lines 428, as in the second illustrative example. However, address and data signals for reading from and writing to the memory are accessed as a specific serial I/O protocol via the TDI-TDO pins, which in this
5 illustrative example are coupled to both of the buses 212 and 214. Because the serial I/O protocol may be sent directly via the TDI-TDO pins, the address register 804 and the data generator 412 are not needed. The serial I/O protocol will vary depending on the size of the serial
10 EEPROM device connected to the access interface 206. For example, for smaller devices the serial address may be 10-bits long, and for larger devices up to 17-bits of serial address may be provided. The clock generator 810 of Fig. 13 provides an access clock signal, SCK, which is divided
15 down from the TCK clock. A programmable clock divider circuit is provided within the clock generator 810 to accommodate a variety of SCK clock frequencies.

A first method of operation of the access interface 206 is illustrated by reference to Figs. 10 and 14. Specifically, the timing diagram of Fig. 10 and the flow
20 diagram of Fig. 14 depict the timing and software flow, respectively, for a typical series of memory access operations, e.g., a read operation, immediately followed by a write operation, and immediately followed by another
25 read operation. Before the memory access operations are performed, the access interface 206 is initialized as follows. First, the modified TAP Controller 402 is reset, as depicted in step 1402 (see Fig. 14), thereby placing the modified TAP Controller 402 in the Test-Logic-Reset
30 502 state. This also causes the AI_RESET signal to be asserted, thereby resetting the access interface 206.

-38-

Next, as depicted in step 1404, a Shift instruction is loaded into the IR 404 for use in scanning in via the TDI pin a suitable read/write/read control protocol into the sequence register 802 and initializing the address register 804 with a starting address. A memory access instruction is then loaded, as depicted in step 1406, into the IR 404 to assert the AI_ENABLE signal, program the Access1-DR 536 state to perform a Shift operation, program the Update1/Capture1-DR 538 state to perform an Update operation and a Capture operation, program the Access2-DR 540 state to perform a Shift operation, program the Update2/Capture2-DR 542 state to inhibit both the Update and Capture operations, and select a register in the data generator 412. Next, the level of TMS is set to logical 0, as depicted in step 1408, to transition the modified TAP Controller 402 FSM first to the Enter-Access-DR 534 state and then to the Access1-DR 536 state (see Figs. 5 and 10). Scan-in data is then shifted, as depicted in step 1410, via the bus 420 into the selected register of the data generator 412 from the memory location pointed to by the starting address; and, a first read control sequence is concurrently applied, as also depicted in step 1410, via the bus 418 to the memory circuit connected to the access interface 206. As depicted in the timing diagram of Fig. 10, four (4) cycles of the TCK clock occur during the first read operation to shift four (4) bits of data into the selected register of the data generator 412 from the memory circuit. The level of TMS is then set, as depicted in step 1412, to logical 1 to transition the modified TAP Controller 402 FSM to the Update1/Capture1-DR 538 state (see Figs. 5 and 10). Next, the shifted data is

updated, as depicted in step 1414, in the selected register so that it can be written to the memory circuit in the next access sequence. It should be noted that the initial scan-out data, which was shifted-out of the data generator 412 at the TDO pin, is unknown data and may be discarded. Next, previously read data is captured, as depicted in step 1415, in the selected register via the bus 420 from the memory circuit. Such previously read data is subsequently shifted out of the data generator 412 via the bus 212. The level of TMS is then set to logical 0, as depicted in step 1416, to transition the modified TAP Controller 402 FSM to the Access2-DR 540 state (see Figs. 5 and 10). Next, the data in the selected register is shifted, as depicted in step 1418, out of the data generator 412 via the bus 420 to the memory; and, the protocol generator 410 concurrently applies, as also depicted in step 1418, a write control sequence via the bus 418 to the memory and an address control sequence via the AR_CNTL signals to the address register 804 to increment the stored address. The level of TMS is then set to logical 1, as depicted in step 1420, to transition the modified TAP Controller 402 FSM to the Update2/Capture2-DR 542 state (see Figs. 5 and 10), thereby ending the write operation. Next, a decision is made, as depicted in step 1422, as to whether the second memory read sequence has been performed. If not, then the method loops back to step 1408, thereby starting the second memory read sequence. Otherwise, the level of TMS is set to logical 1 and then to logical 0, as depicted in step 1424, to transition the modified TAP Controller 402 FSM first to the Exit-Access-DR 544 state and then to the

Run-Test/Idle 504 state (see Figs. 5 and 10), thereby ending the read/write/read sequence. It should be noted that the above-described method may be suitably modified for performing any desired number of memory read/write
5 operations.

A second method of operation of the access interface 206 is illustrated by reference to Figs. 11 and 15. Specifically, the timing diagram of Fig. 11 and the flow diagram of Fig. 15 depict the timing and software flow,
10 respectively, for a memory access sequence that scans in an initial data value into the data generator 412, executes a control protocol sequence using the control queue of the sequence register 802 that repeatedly executes a sequence of memory read/write operations, and
15 finally scans out a data result using the data generator 412. This exemplary sequence may be used to execute a memory BIST algorithm. First, the modified TAP Controller 402 is reset, as depicted in step 1502 (see Fig. 15), to place the modified TAP Controller 402 in the Test-Logic-
20 Reset 502 state. This also causes the AI_RESET signal to be asserted, thereby resetting the access interface 206. Next, a Shift instruction is loaded, as depicted in step 1504, into the IR 404 for use in loading the control queue of the sequence register 802 with the BIST control
25 protocol and initializing the address register 804 with a starting address. A memory BIST instruction is then loaded, as depicted in step 1506, into the IR 404 to set the AI_ENABLE signal high, program the Access1-DR 536 state to perform a Shift operation, program the
30 Update1/Capture1-DR 538 state to perform an Update operation, program the Access2-DR 540 to perform a memory

BIST sequence, program the Update2/Capture2-DR 542 state to perform a Capture operation, and select a register in the data generator 412. The memory BIST sequence performed by the Access2-DR 540 will be described in detail below. Next, the level of TMS is set to logical 0, as depicted in step 1508, to transition the modified TAP Controller 402 FSM first to the Enter-Access-DR 534 state and then to the Access1-DR 536 state (see Figs. 5 and 11). Scan-in data is then shifted, as depicted in step 1510, via the TDI pin into the selected register of the data generator 412 for use in initializing the register for a first BIST sequence. It should be noted that in this exemplary method, a memory control sequence does not start concurrently with the data generator 412 Shift operation.

Next, the level of TMS is set to logical 1, as depicted in step 1512, to transition the modified TAP Controller 402 FSM to the Update1/Capture1-DR 538 state (see Figs. 5 and 11). The shifted data is then updated, as depicted in step 1514, in the selected register. Next, the level of TMS is then set to logical 0, as depicted in step 1516, to transition the modified TAP Controller 402 FSM to the Access2-DR 540 state (see Figs. 5 and 11) to execute the memory BIST sequence. This may take many cycles of the TCK clock to complete. Specifically, in a typical memory BIST sequence, the data stored in the selected register is written to a memory location, read back from that location, and then written to and read back from successive memory locations until the test is completed. Accordingly, the address stored in the address register 804 is suitably incremented (or decremented), as further depicted in step 1516, and the sequence register 802 is

sequenced and looped, as still further depicted in step 1516, on memory read/write commands based upon the programming performed in steps 1504 and 1506. Next, the level of TMS is set to logical 1, as depicted in step 5 1518, to transition the modified TAP Controller 402 FSM to the Update2/Capture2-DR 542 state. The result of the memory BIST sequence is then captured, as depicted in step 1519, in the selected register (see Figs. 5 and 10). A decision is then made, as depicted in step 1520, as to whether there is any more data to scan into the selected register of the data generator 412. If so, then the method loops back to step 1508. As depicted in Fig. 11, there is valid data at both the TDI and TDO pins during the subsequent execution of the step 1510. The valid TDO 15 data is the result of the first memory BIST sequence, and the valid TDI data is new scan data to be used in a next memory BIST sequence. Otherwise, if there is no more data to be scanned into the data generator 412, then another decision is made, as depicted in step 1522, as to whether 20 the control queue of the sequence register 802 is to be loaded with another BIST control protocol. If so, then the method loops back to step 1504. Otherwise, the level of TMS is first set to logical 1 and then to logical 0, as depicted in step 1524, to transition the modified TAP 25 Controller 402 FSM first to the Exit-Access-DR 544 state and then to the Run-Test/Idle 504 state (see Figs. 5), thereby ending the memory BIST sequence. In this exemplary method, the memory BIST sequence may be repeated with new data in the selected register, or the modified 30 TAP Controller 402 FSM may transition to the Exit-Access-DR 544 state and the sequence register 802 and the address

register 804 re-initialized to start a new memory BIST sequence.

Having described the above illustrative embodiments, other alternative embodiments or variations may be made. Specifically, Figs. 6 and 7 depict two (2) representative implementations of the data generator 412. However, the access interface 206 is intended to support many different implementations of the data generator 412, depending upon the specific application, e.g., programming, testing, or debugging, and the design and type of circuit connected to it.

Specifically, it was mentioned above that it is possible for the data generator 412 to operate as a pattern generator, a comparator, or a signature register in order to support BIST of a circuit coupled to the access interface 206. For example, in a BIST implementation of the data generator 412, the Shift/From register 702 (see Fig. 7) may be designed to function as an MISR and the Shift/To register 704 (see Fig. 7) may be designed to function as an LFSR. Further, although two (2) separate scan paths are shown in Fig. 7, these scan paths may be connected to form a single scan path as depicted in Fig. 7 by the dashed line connecting TDO_1 and TDI_2. In this application, the TDO_1 output and TDI_2 input are not included in the data generator 412, and there is only a single TDI-TDO path, as depicted in Fig. 6.

It should also be noted that the data generator 412 allows for asymmetrical data to be sent to and received from the connected circuit. For example, the single register data generator 412, as depicted in Fig. 6, may be

connected to a circuit in which the number of outputs is not equal to the number of inputs. In this application, the shift-out/shift-in data for Capture and Update operations may be appropriately "padded" or masked.

5 In addition, it was described that the TAP Controller 402 is preferably a modified TAP Controller that not only includes the standard Select-DR and Select-IR branches 501 and 503, but also includes a new access interface branch 505 with states for implementing operations using the
10 access interface 206. However, this was merely an illustrative example. The TAP Controller 402 FSM may alternatively consist of the standard FSM, as depicted in Fig. 3. The access interface 206 may therefore optionally include decode logic 490 (see Fig. 4) for decoding the
15 states included in the access interface branch 505. Specifically, the decode logic 490 receives the standard FSM states and the logic level on the TMS pin as inputs; and, provides DR_CNTL2 signals in place of the DR_CNTL signals to control register operations.

20 Those of ordinary skill in this art will also appreciate that variations to and modification of the above-described apparatus and method for accessing electrical nodes of electronic circuits may be made without departing from the inventive concepts disclosed herein. Accordingly,
25 the present invention should be viewed as limited solely by the scope and spirit of the appended claims.

CLAIMS

What is claimed is:

1. An interface for accessing electrical nodes of an
5 electronic circuit, comprising:
 a serial input terminal;
 a serial output terminal;
 a control output bus for communicably coupling the
interface to the electronic circuit; and
10 a protocol generator operatively connected between the
serial input terminal and the serial output terminal, the
protocol generator including a control register queue having
a serial input coupled to the serial input terminal for
receiving at least one control sequence into the control
15 register queue and a serial output coupled to the serial
output terminal for forwarding the control sequence from the
control register queue, the control register queue further
including a plurality of control outputs coupled to the
control output bus for providing in parallel the at least
20 one control sequence to the electronic circuit.
2. The interface of claim 1 wherein the control register
queue includes a plurality of control registers, each
control register being serially connected to at least one
25 control register adjacent thereto to form a scan path.
3. The interface of claim 1 further including an address
bus for communicably coupling the interface to the
electronic circuit, and wherein the protocol generator
30 further includes at least one address register connected
between the serial input terminal and the serial output

terminal for receiving at least one address sequence into the address register and for forwarding the address sequence from the address register, the address register including a plurality of address outputs coupled to the address bus for providing in parallel the at least one address sequence to the electronic circuit.

4. The interface of claim 3 wherein the at least one address register comprises a plurality of address registers, each address register being serially connected to at least one address register adjacent thereto to form a scan path.

5. An interface for accessing electrical nodes of an electronic circuit, comprising:

- a serial input terminal;
- a serial output terminal;
- a data bus for communicably coupling the interface to the electronic circuit; and
- a data generator operatively connected between the serial input terminal and the serial output terminal, the data generator including at least one data register having a serial input coupled to the serial input terminal for receiving at least one data sequence into the data register and a serial output coupled to the serial output terminal for forwarding the data sequence from the data register, the data register further including a plurality of data terminals coupled to the data bus for providing in parallel the at least one data sequence to the electronic circuit and for receiving in parallel data from the electronic circuit.

6. The interface of claim 5 wherein the data generator includes a plurality of data registers, each data register being serially connected to at least one data register adjacent thereto to form a scan path.

5

7. A digital controller operatively connectable to an interface for accessing electrical nodes of an electronic circuit, the interface having at least one scan path for applying digital information to and receiving digital
10 information from the electronic circuit, the digital controller comprising:

a state machine operatively connected to the at least one scan path and having a plurality of states, the interface performing operational commands based on the
15 plurality of states, and at least one of the plurality of states being programmable.

8. The digital controller of claim 7 wherein the at least one programmable state is a programmed state selected from
20 the group consisting of a shift-data state and a pause state.

9. The digital controller of claim 7 wherein the at least one programmable state is a programmed state selected from
25 the group consisting of an update-data state and a capture-data state.

10. The digital controller of claim 7 wherein the at least one programmable state is a sequence of states programmed to
30 perform an update operation followed by a capture operation.

11. The digital controller of claim 7 wherein the at least one programmable state is a sequence of states programmed to perform a shift operation followed by a capture operation.

5 12. The digital controller of claim 11 wherein the sequence of states is programmed to be performed a predetermined number of times.

10 13. The digital controller of claim 7 wherein the at least one programmable state is a sequence of states programmed to perform a capture operation followed by a shift operation.

15 14. The digital controller of claim 13 wherein the sequence of states is programmed to be performed a predetermined number of times.

15. An interface for accessing electrical nodes of an electronic circuit, comprising:

- 20 a serial input terminal;
- a serial output terminal;
- a control output bus for communicably coupling the interface to the electronic circuit;
- a data bus for communicably coupling the interface to the electronic circuit;
- 25 an address bus for communicably coupling the interface to the electronic circuit;
- a protocol generator operatively connected between the serial input terminal and the serial output terminal, the protocol generator including a plurality of first control
- 30 outputs coupled to the control output bus and a plurality of address outputs coupled to the address bus, the protocol

generator for performing operations including providing in parallel at least one control sequence to the electronic circuit using the plurality of first control outputs and providing in parallel at least one address sequence to the electronic circuit using the plurality of address outputs;
5 and

a data generator operatively connected between the serial input terminal and the serial output terminal and including a plurality of data terminals coupled to the data bus, the data generator for performing operations including receiving in parallel data from the electronic circuit and providing in parallel at least one data sequence to the electronic circuit using the plurality of data terminals, the receiving and providing operations of the data generator
10 being performed concurrently with the providing operations of the protocol generator.

16. The interface of claim 15 wherein the protocol generator further includes at least one second control output communicably coupled to the data generator for
20 controlling the provision of the data sequence to the electronic circuit.

17. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:
25

coupling a digital controller to an interface for accessing electrical nodes of the electronic circuit, the digital controller including a state machine connected to at least one scan path of the interface and having a plurality
30 of states;

in response to a predetermined event, programming at least one of the plurality of states so as to program a corresponding operational command performed by the interface based thereon; and

- 5 performing the programmed operational command using the interface to access at least one electrical node of the electronic circuit.

18. A method of accessing electrical nodes of an electronic circuit, used with an interface operatively coupled to the electronic circuit and including a control register queue having a first serial input and a plurality of parallel outputs operatively coupled to the electronic circuit, at least one data register having a second serial input and a plurality of data terminals operatively coupled to the electronic circuit, and at least one address register having a third serial input and a plurality of address terminals operatively coupled to the electronic circuit, the method comprising the steps of:

- 20 scanning-in at least one first control sequence into the control register queue using the first serial input;

scanning-in at least one first data sequence into the at least one data register using the second serial input;

- 25 scanning-in at least one first address sequence into the at least one address register using the third serial input;

and

- 30 concurrently applying the at least one first control sequence to the electronic circuit using the plurality of parallel outputs, the at least one first data sequence to the electronic circuit using the plurality of data

terminals, and the at least one first address sequence to the electronic circuit using the plurality of address terminals to access at least one first electrical node of the electronic circuit.

5

19. The method of claim 18 further including the steps of scanning-in at least one second control sequence into the control register queue using the first serial input;

scanning-in at least one second address sequence into
10 the at least one address register using the third serial input; and

concurrently applying the at least one second control sequence to the electronic circuit using the plurality of parallel outputs and the at least one second address
15 sequence to the electronic circuit using the plurality of address terminals, and receiving at least one second data sequence from the electronic circuit using the plurality of data terminals to access at least one second electrical node of the electronic circuit.

20

20. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:

programming a programmable logic device with a representation of the interface of claim 1;

25 operatively coupling the programmed logic device to the electronic circuit; and

accessing at least one electrical node of the electronic circuit using the programmed logic device.

30 21. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:

-52-

programming a programmable logic device with a representation of the interface of claim 5;

operatively coupling the programmed logic device to the electronic circuit; and

5 accessing at least one electrical node of the electronic circuit using the programmed logic device.

22. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:

10 programming a programmable logic device with a representation of the interface of claim 15;

operatively coupling the programmed logic device to the electronic circuit; and

15 accessing at least one electrical node of the electronic circuit using the programmed logic device.

23. An apparatus communicably coupleable to the interface of claim 1 and operative to control the interface for accessing electrical nodes of an electronic circuit.

20

24. An apparatus communicably coupleable to the interface of claim 5 and operative to control the interface for accessing electrical nodes of an electronic circuit.

25 25. An apparatus communicably coupleable to the interface of claim 15 and operative to control the interface for accessing electrical nodes of an electronic circuit.

26. The apparatus of claim 23 wherein the apparatus
30 comprises a test resource.

27. The apparatus of claim 23 wherein the apparatus comprises a programming resource.

28. The apparatus of claim 23 wherein the apparatus
5 comprises a debugging resource.

29. The apparatus of claim 23 wherein the apparatus and the interface are implemented on an integrated circuit.

10 30. The apparatus of claim 23 wherein the interface is implemented on an integrated circuit and the apparatus is external to the integrated circuit.

AMENDED CLAIMS

[received by the International Bureau on 15 August 2001 (15.08.01);
original claims 20-22 amended; remaining claims unchanged (4 pages)]

terminals, and the at least one first address sequence to the electronic circuit using the plurality of address terminals to access at least one first electrical node of the electronic circuit.

5

19. The method of claim 18 further including the steps of scanning-in at least one second control sequence into the control register queue using the first serial input;

scanning-in at least one second address sequence into
10 the at least one address register using the third serial input; and

concurrently applying the at least one second control sequence to the electronic circuit using the plurality of parallel outputs and the at least one second address
15 sequence to the electronic circuit using the plurality of address terminals, and receiving at least one second data sequence from the electronic circuit using the plurality of data terminals to access at least one second electrical node of the electronic circuit.

20

20. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:

programming a programmable logic device with a representation of an interface circuit comprising

25 a serial input terminal,
a serial output terminal,
a control output bus for communicably coupling the interface to the electronic circuit, and

30 a protocol generator operatively connected between the serial input terminal and the serial output terminal, the protocol generator including a control register queue having

a serial input coupled to the serial input terminal for receiving at least one control sequence into the control register queue and a serial output coupled to the serial output terminal for forwarding the control sequence from the control register queue, the control register queue further including a plurality of control outputs coupled to the control output bus for providing in parallel the at least one control sequence to the electronic circuit;

operatively coupling the programmed logic device to the electronic circuit; and

accessing at least one electrical node of the electronic circuit using the programmed logic device.

21. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:

programming a programmable logic device with a representation of an interface circuit comprising

a serial input terminal,

a serial output terminal,

a data bus for communicably coupling the interface to the electronic circuit, and

a data generator operatively connected between the serial input terminal and the serial output terminal, the data generator including at least one data register having a serial input coupled to the serial input terminal for receiving at least one data sequence into the data register and a serial output coupled to the serial output terminal for forwarding the data sequence from the data register, the data register further including a plurality of data terminals coupled to the data bus for providing in parallel

the at least one data sequence to the electronic circuit and for receiving in parallel data from the electronic circuit;

operatively coupling the programmed logic device to the electronic circuit; and

5 accessing at least one electrical node of the electronic circuit using the programmed logic device.

22. A method of accessing electrical nodes of an electronic circuit, comprising the steps of:

10 programming a programmable logic device with a representation of an interface circuit comprising

 a serial input terminal,

 a serial output terminal,

15 a control output bus for communicably coupling the interface to the electronic circuit,

 a data bus for communicably coupling the interface to the electronic circuit,

 an address bus for communicably coupling the interface to the electronic circuit,

20 a protocol generator operatively connected between the serial input terminal and the serial output terminal, the protocol generator including a plurality of first control outputs coupled to the control output bus and a plurality of address outputs coupled to the address bus, the protocol
25 generator for performing operations including providing in parallel at least one control sequence to the electronic circuit using the plurality of first control outputs and providing in parallel at least one address sequence to the electronic circuit using the plurality of address outputs,
30 and

a data generator operatively connected between the serial input terminal and the serial output terminal and including a plurality of data terminals coupled to the data bus, the data generator for performing operations including receiving in parallel data from the electronic circuit and providing in parallel at least one data sequence to the electronic circuit using the plurality of data terminals, the receiving and providing operations of the data generator being performed concurrently with the providing operations of the protocol generator;

operatively coupling the programmed logic device to the electronic circuit; and

accessing at least one electrical node of the electronic circuit using the programmed logic device.

15

23. An apparatus communicably coupleable to the interface of claim 1 and operative to control the interface for accessing electrical nodes of an electronic circuit.

20

24. An apparatus communicably coupleable to the interface of claim 5 and operative to control the interface for accessing electrical nodes of an electronic circuit.

25

25. An apparatus communicably coupleable to the interface of claim 15 and operative to control the interface for accessing electrical nodes of an electronic circuit.

30

26. The apparatus of claim 23 wherein the apparatus comprises a test resource.

1/15

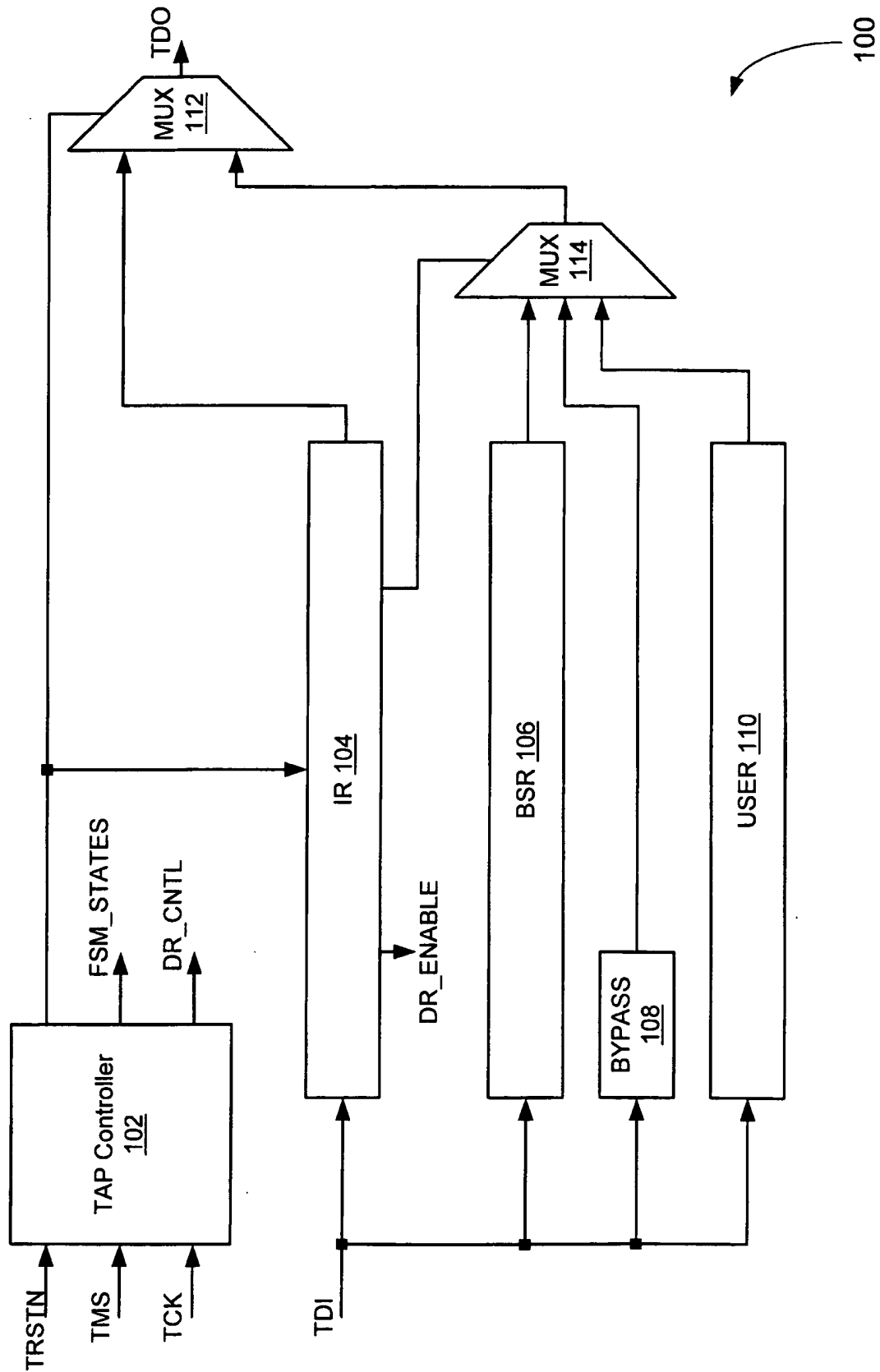


Fig. 1 - Prior Art

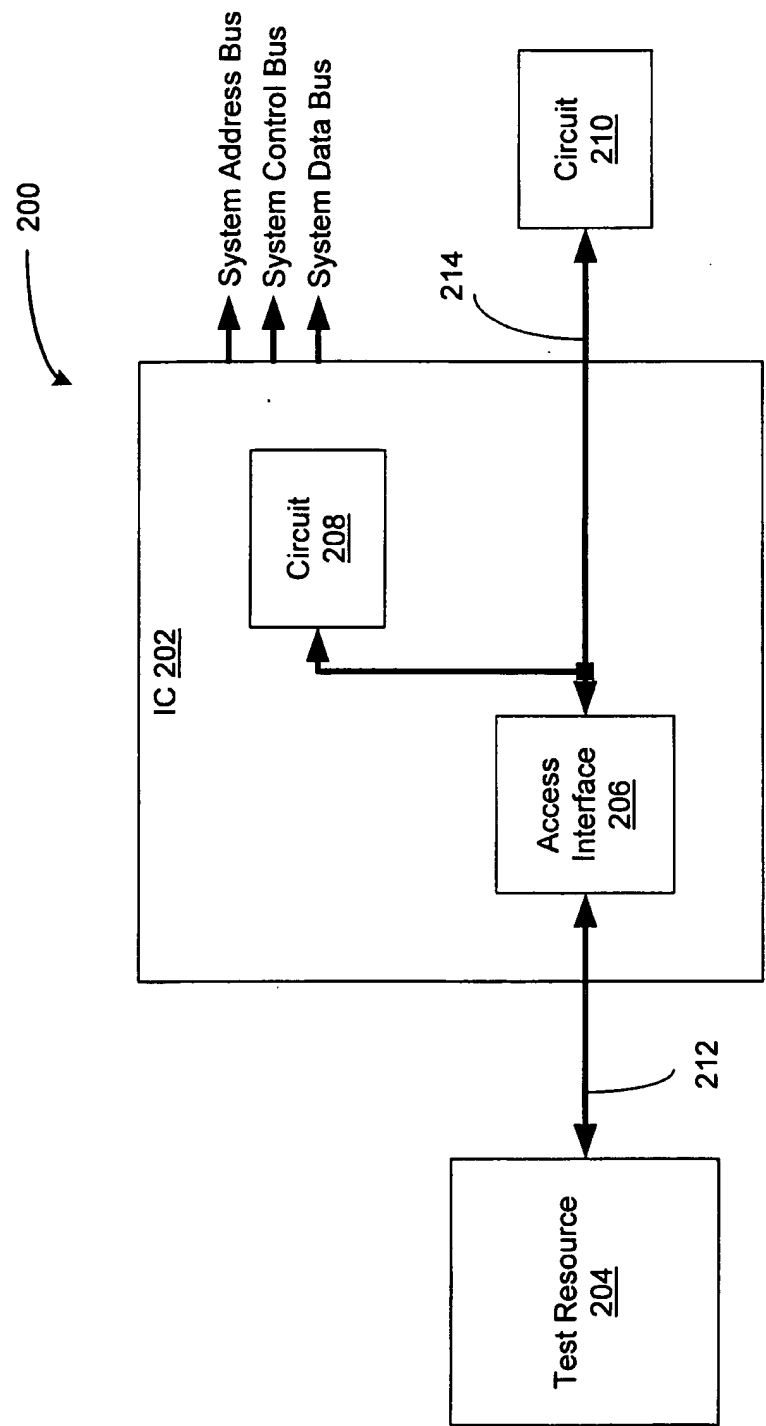


Fig. 2

3/15

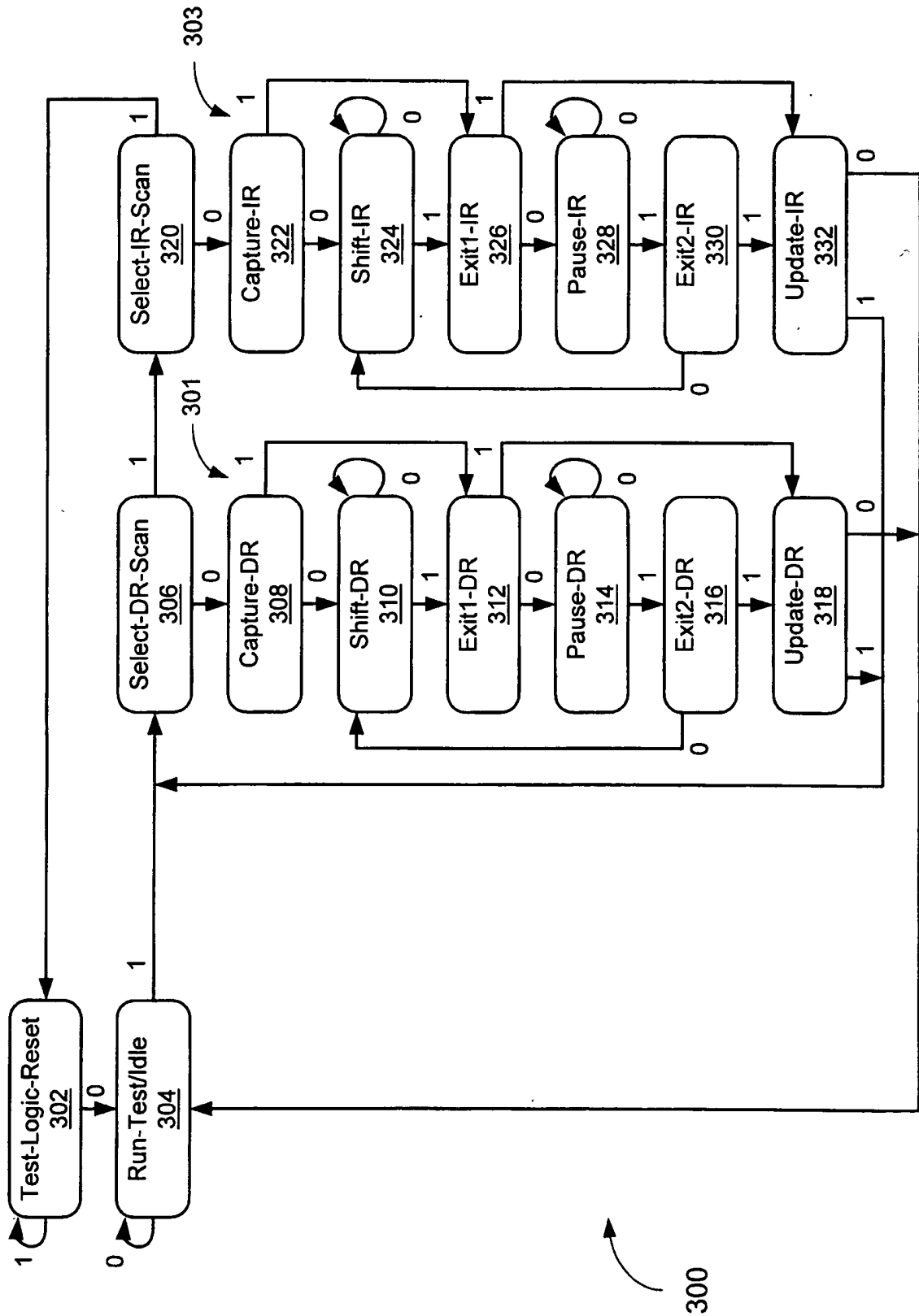


Fig. 3 - Prior Art

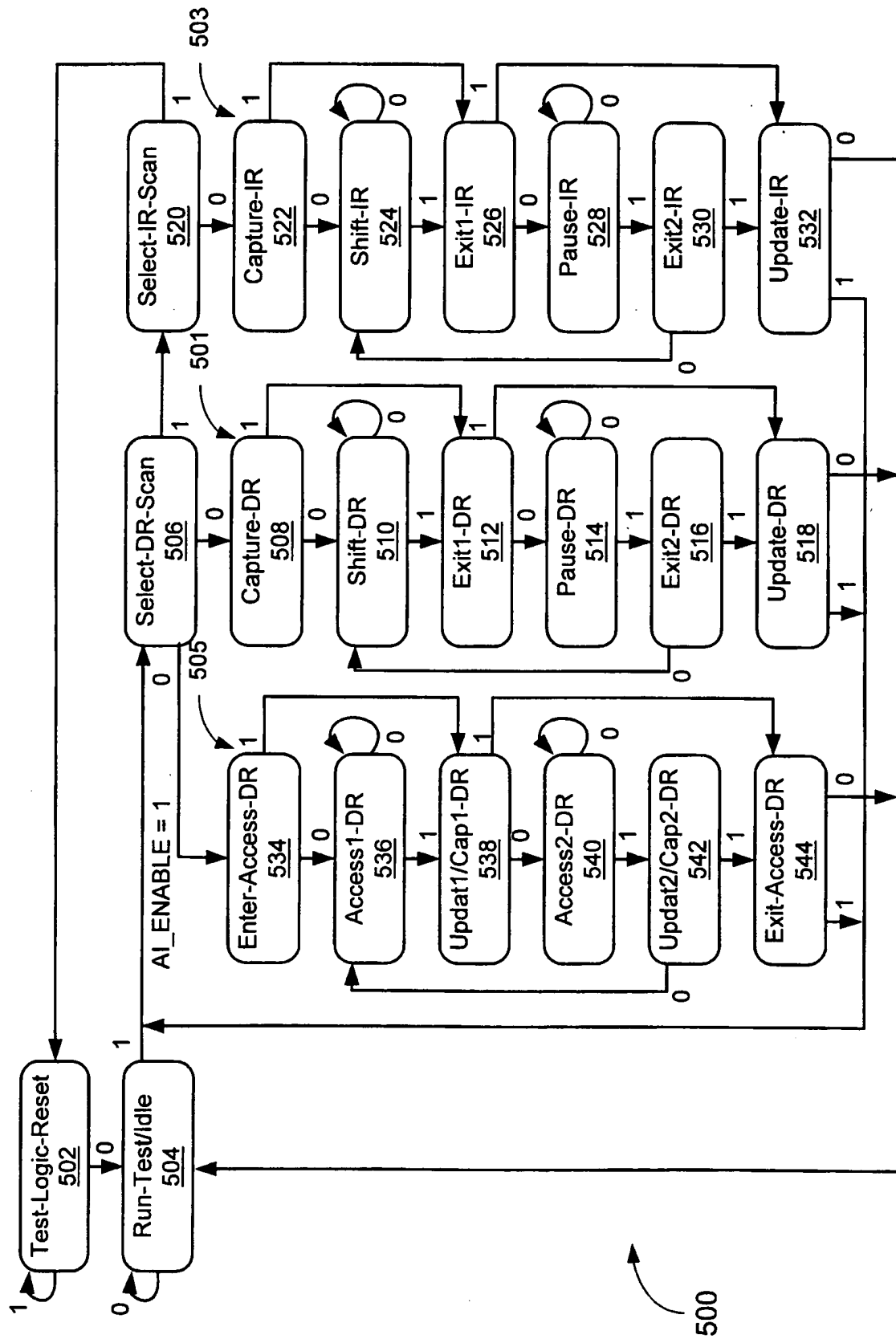


Fig. 5

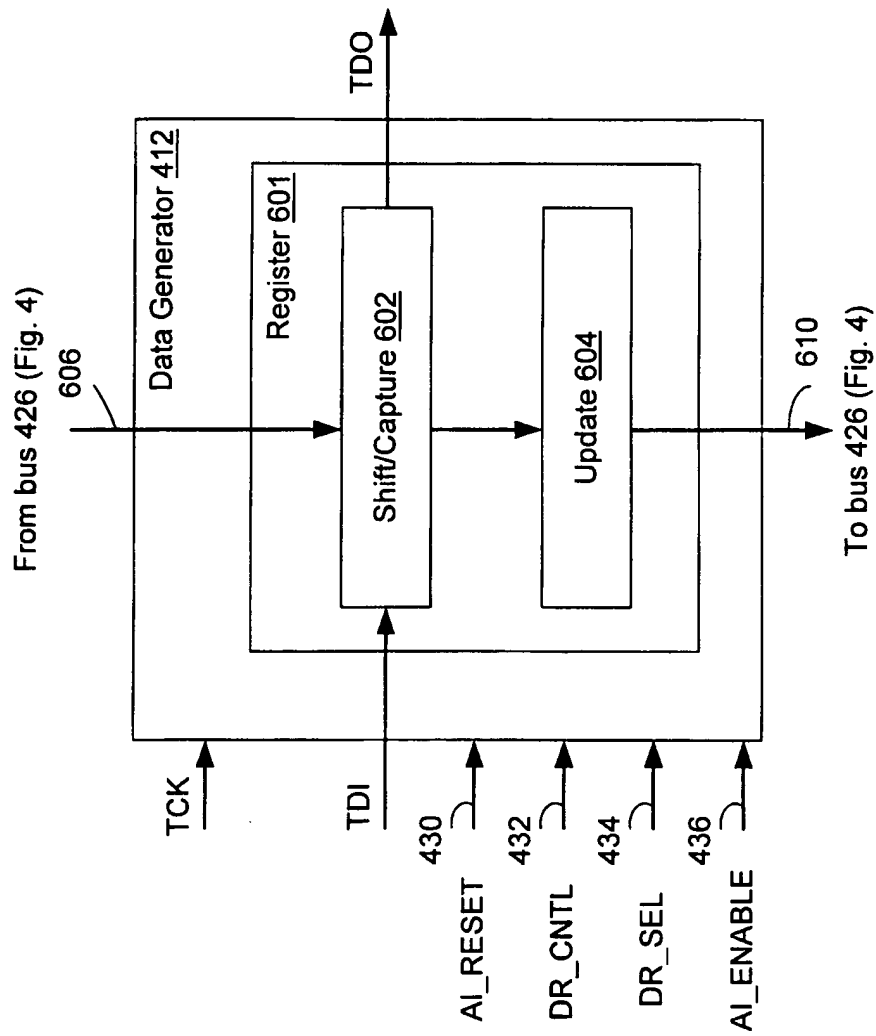
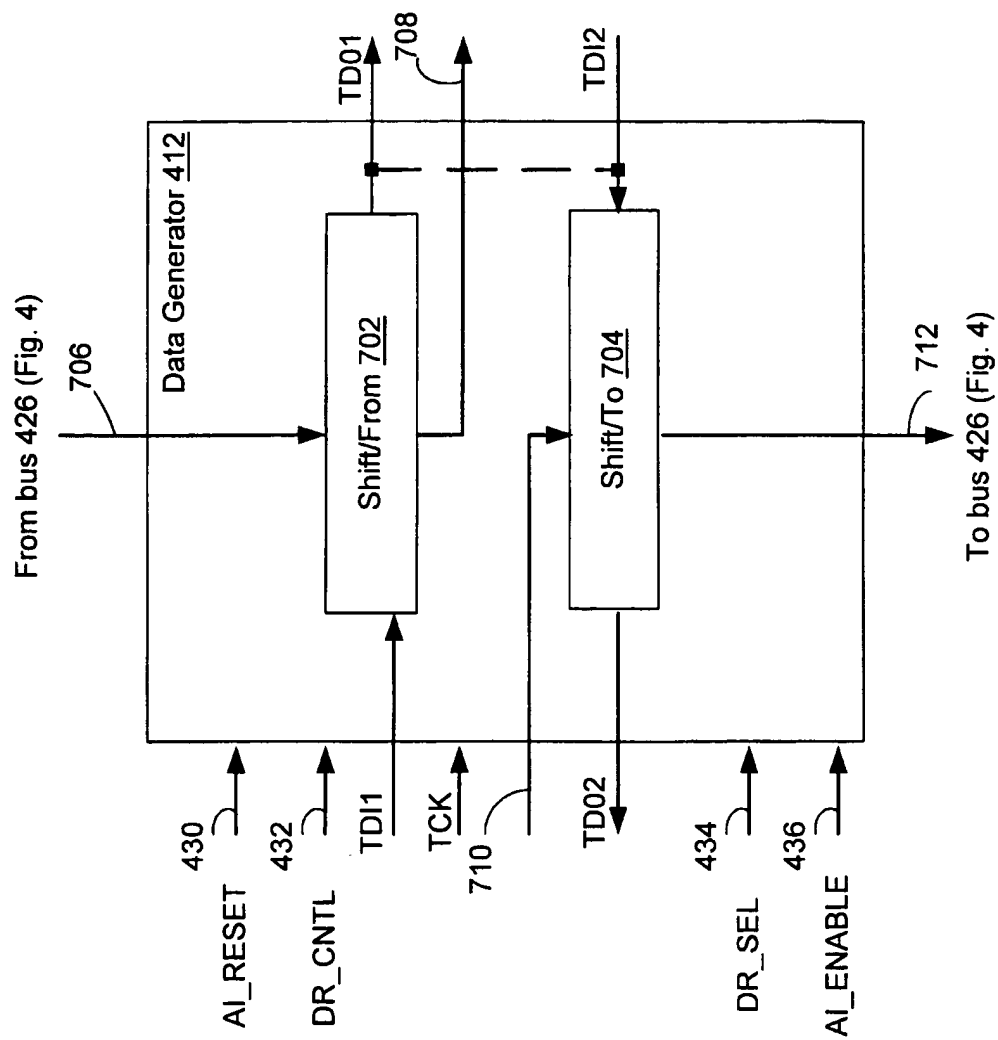


Fig. 6

7/15



8/15

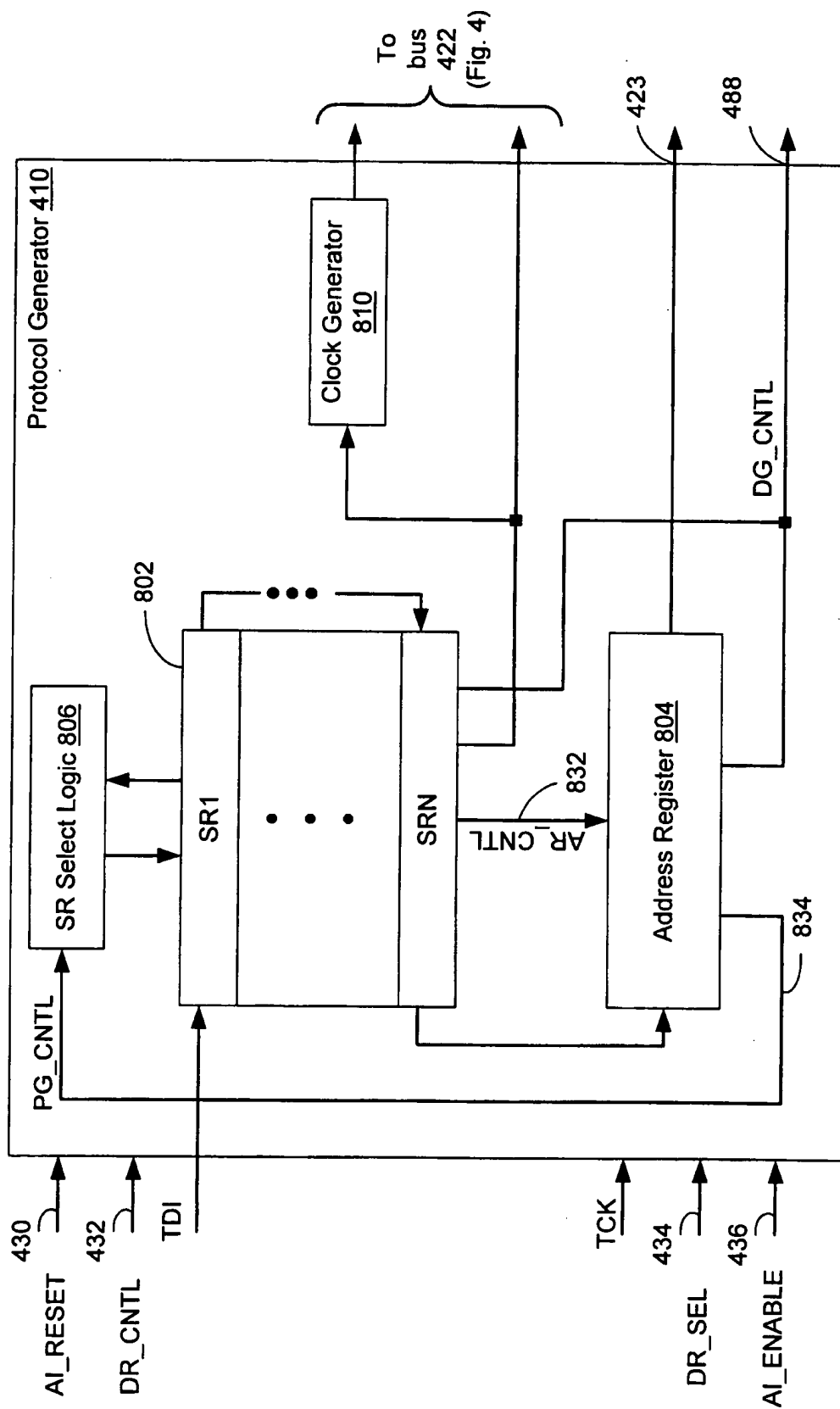


Fig. 8

9/15

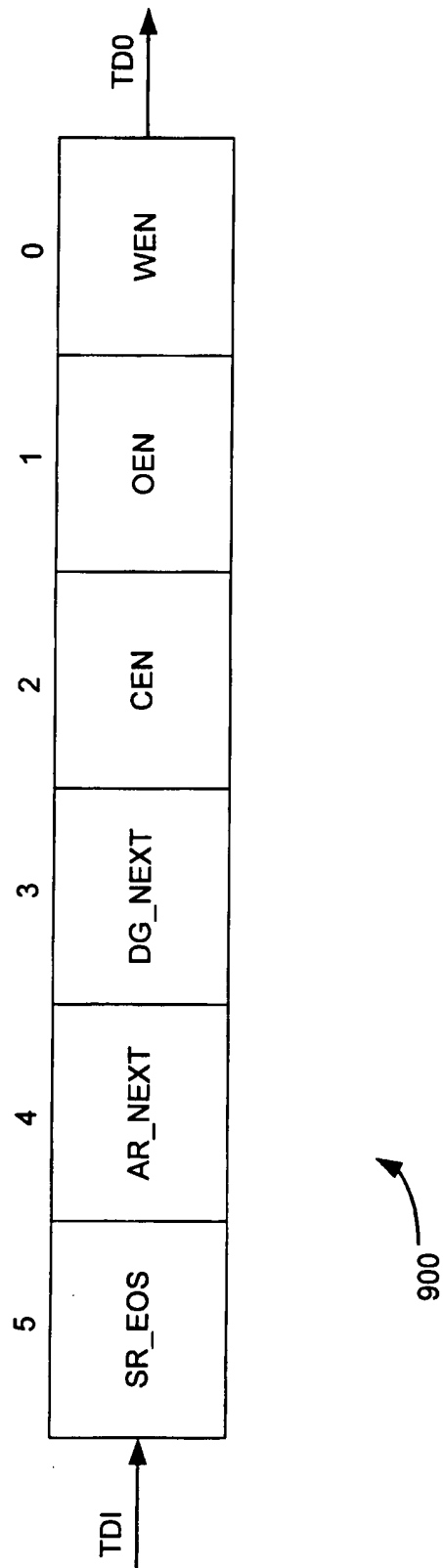


Fig. 9

10/15

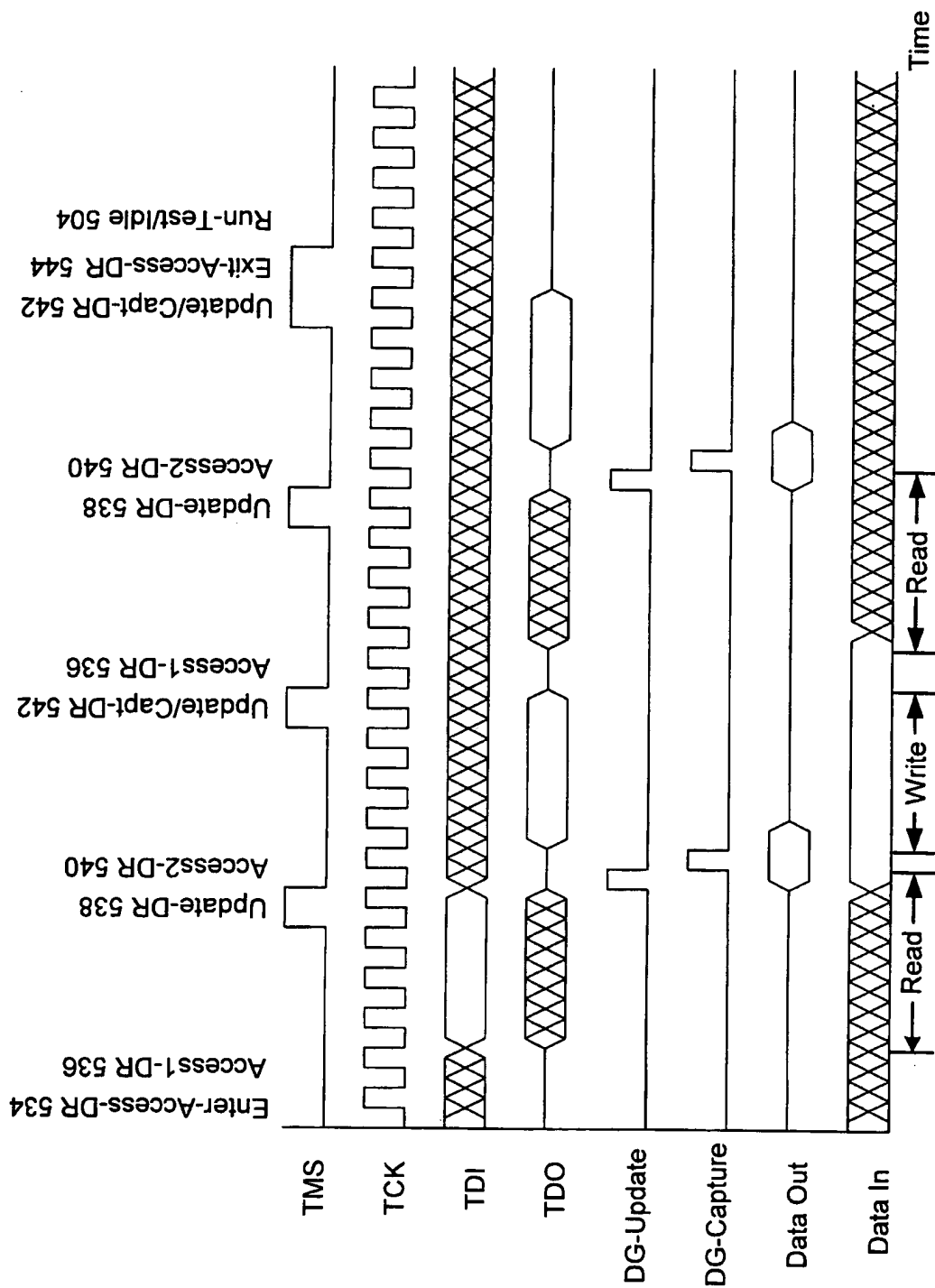


Fig. 10

11/15

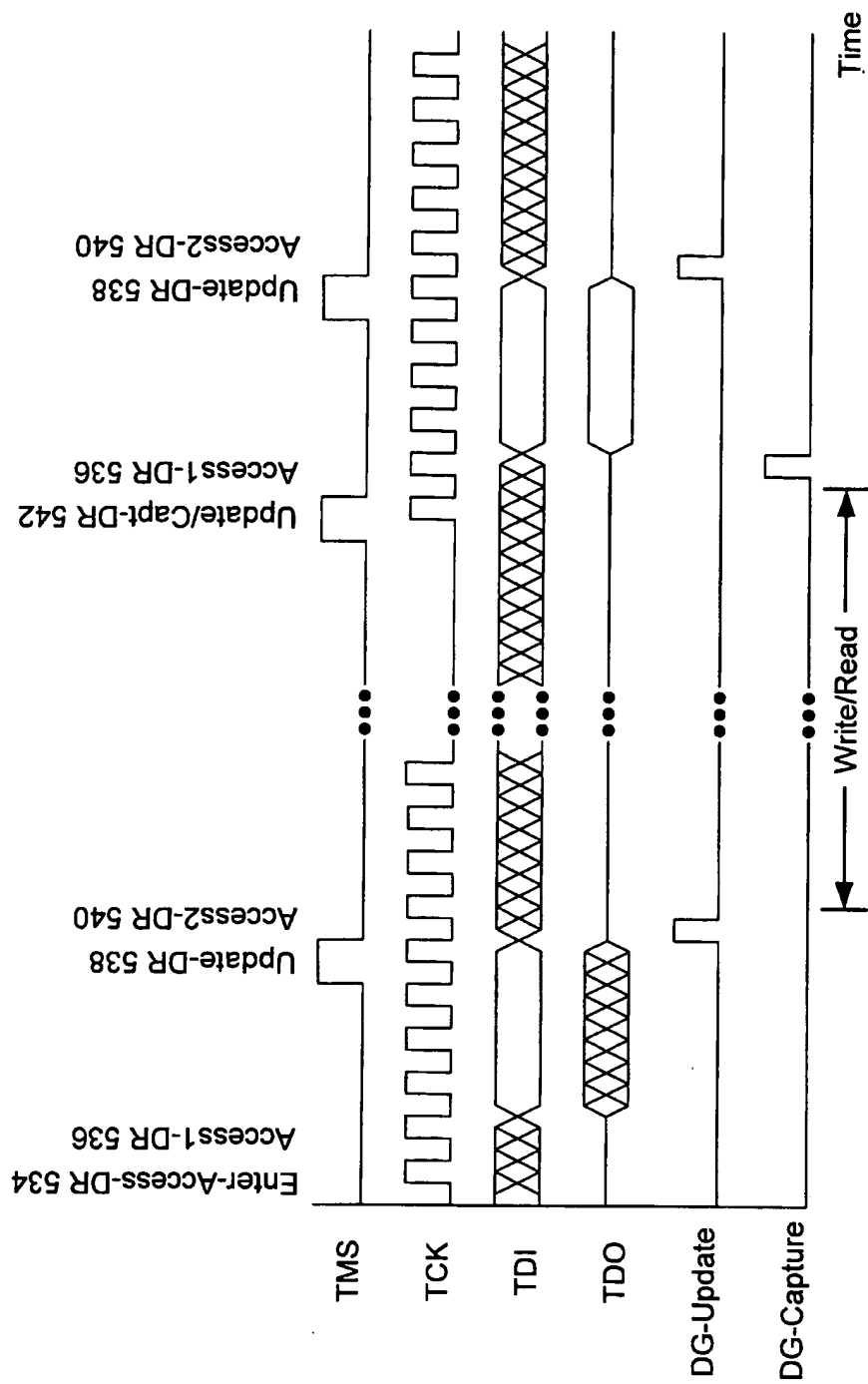
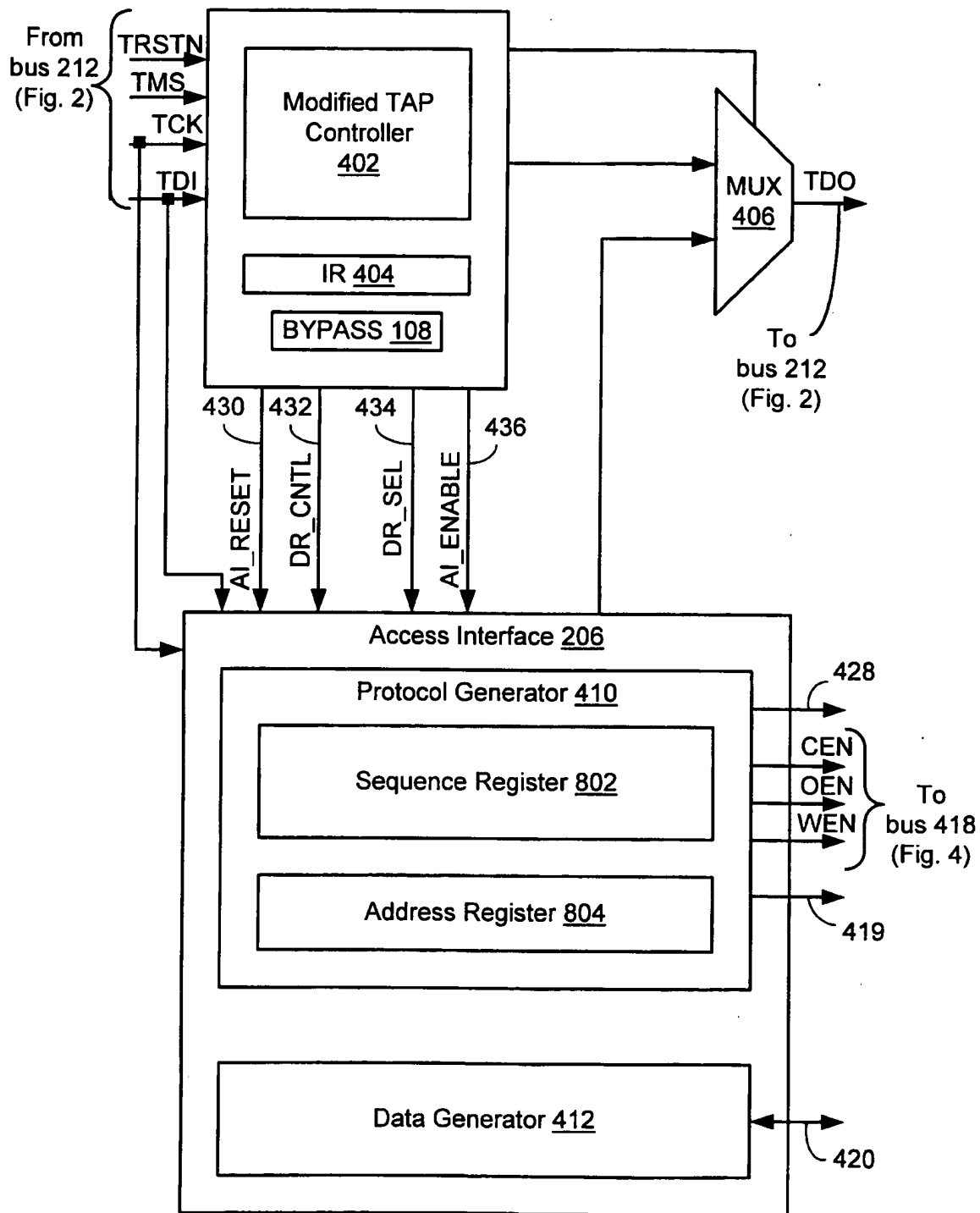
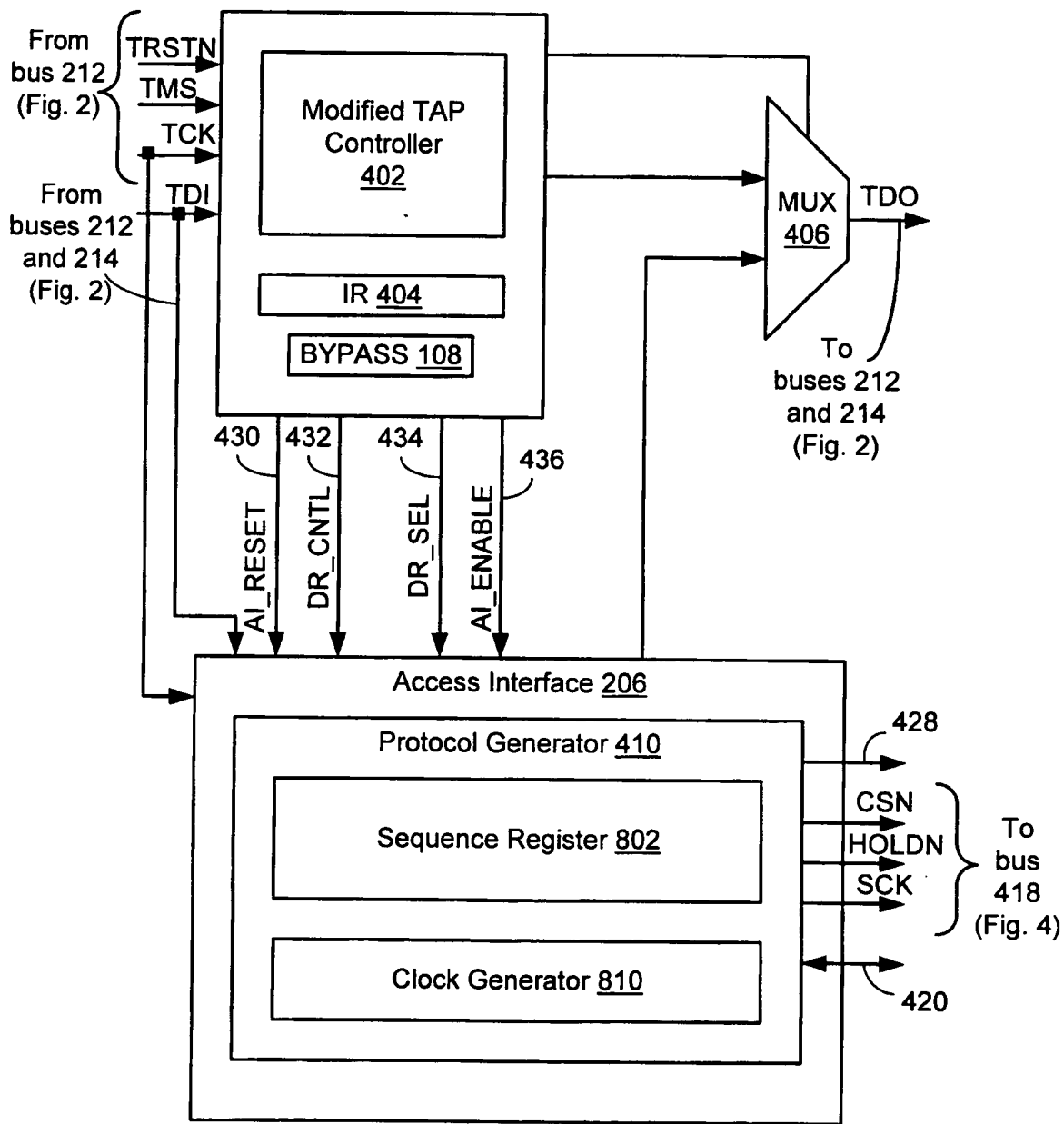


Fig. 11

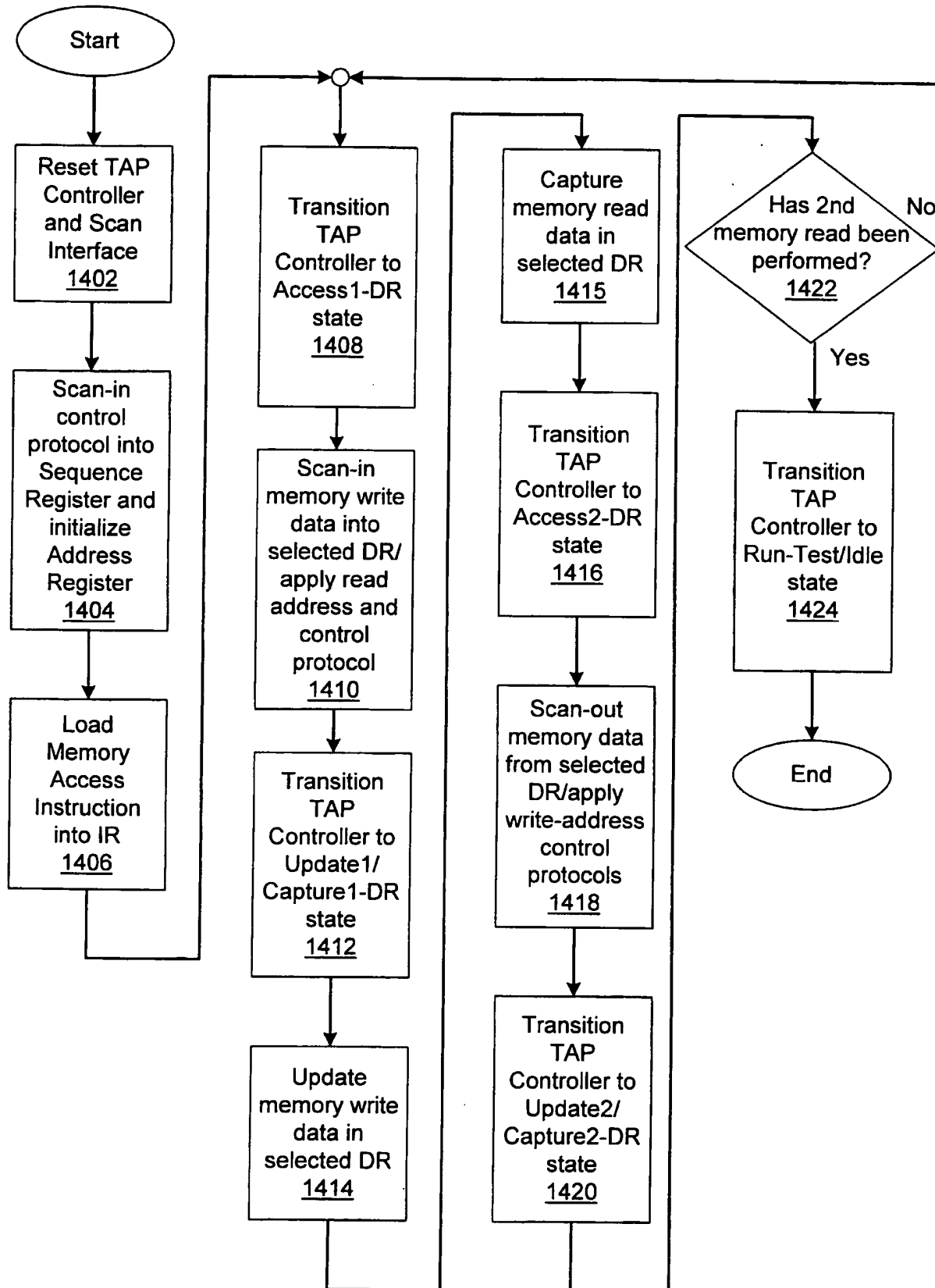
12/15

**Fig. 12**

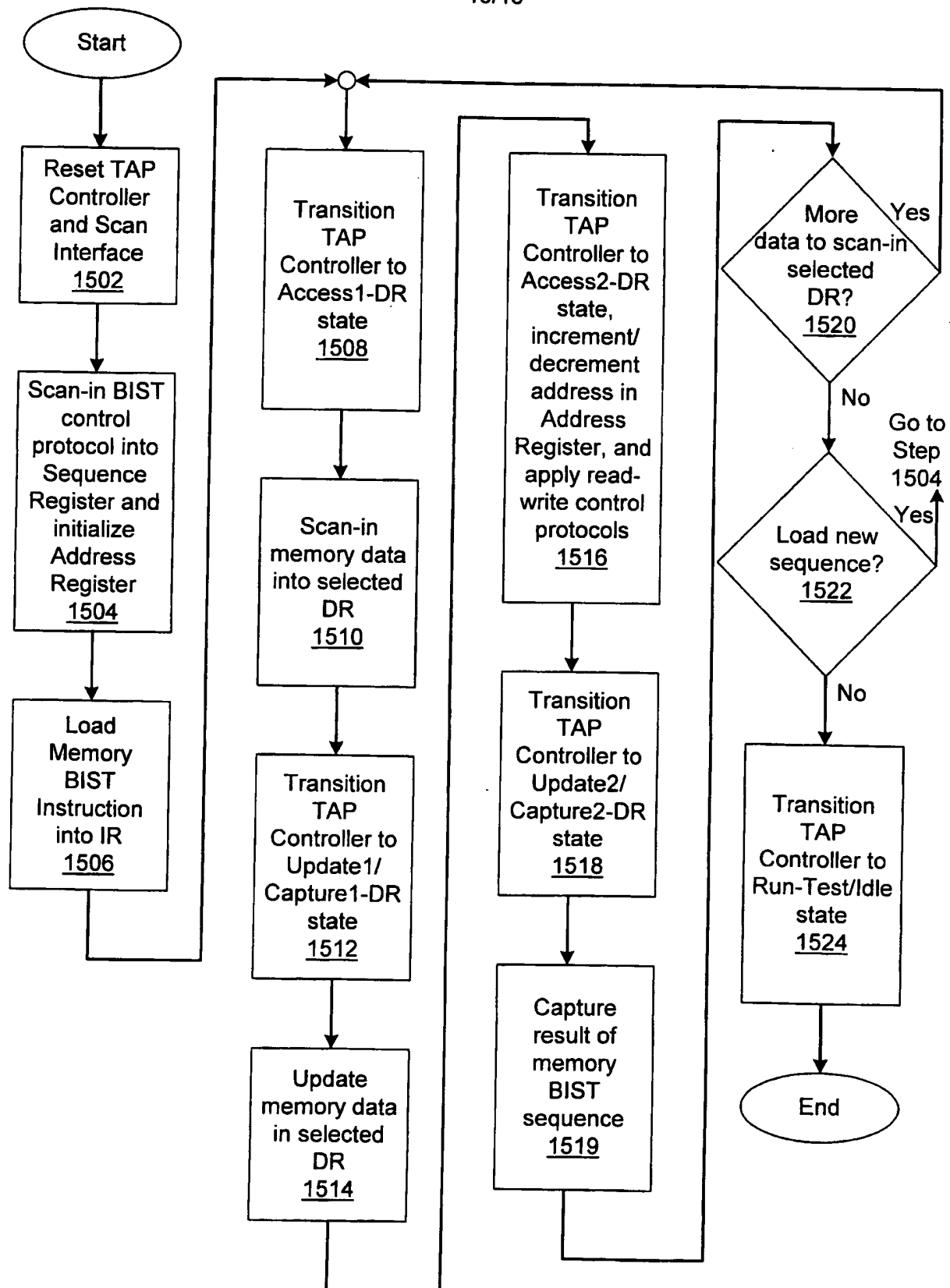
13/15

**Fig. 13**

14/15

**Fig. 14**

15/15

**Fig. 15**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/08884

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H02H 3/05; H03K 19/003

US CL : 714/25, 32, 37, 724, 725, 726; 712/227.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 714/25, 32, 37, 724, 725, 726; 712/227.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

STN, EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,495,486 A (GHEEWALA) 27 February 1996, see abstract, col. 2, lines 37-58; col.7, lines 38-58.	1- 6, 17, 23-30.
Y	US 5,684,721 A (SWOBODA et al) 04 November 1997, see abstract, col.2, lines 44- 65; col.3, lines 5-58.	7-14.
Y	US 6,016,555 A (DEAO et al) 18 January 2000, see abstract, col.5, lines 10-65.	7-14.
A	US 5,627,840 (HUNDERTMARK et al) 06 May 1997, see the entire patent.	1-19, 23-30.
A	US 5,231,314 A (ANDREWS) 27 July 1993, see the entire patent.	1-19, 23-30.

☒ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
B earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Z* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

16 MAY 2001

Date of mailing of the international search report

15 JUN 2001

 Name and mailing address of the ISA/US
 Commissioner of Patents and Trademarks
 Box PCT
 Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

DIEU-MINH LE

Telephone No. (703) 305-9408

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/08884

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,357,572 A [BIANCO et al] 18 October 1994, see the entire patent.	1-19, 23-30.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/08884

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☒ Claims Nos.: 20, 21, 22
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐

The additional search fees were accompanied by the applicant's protest.

☐

No protest accompanied the payment of additional search fees.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☒ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.